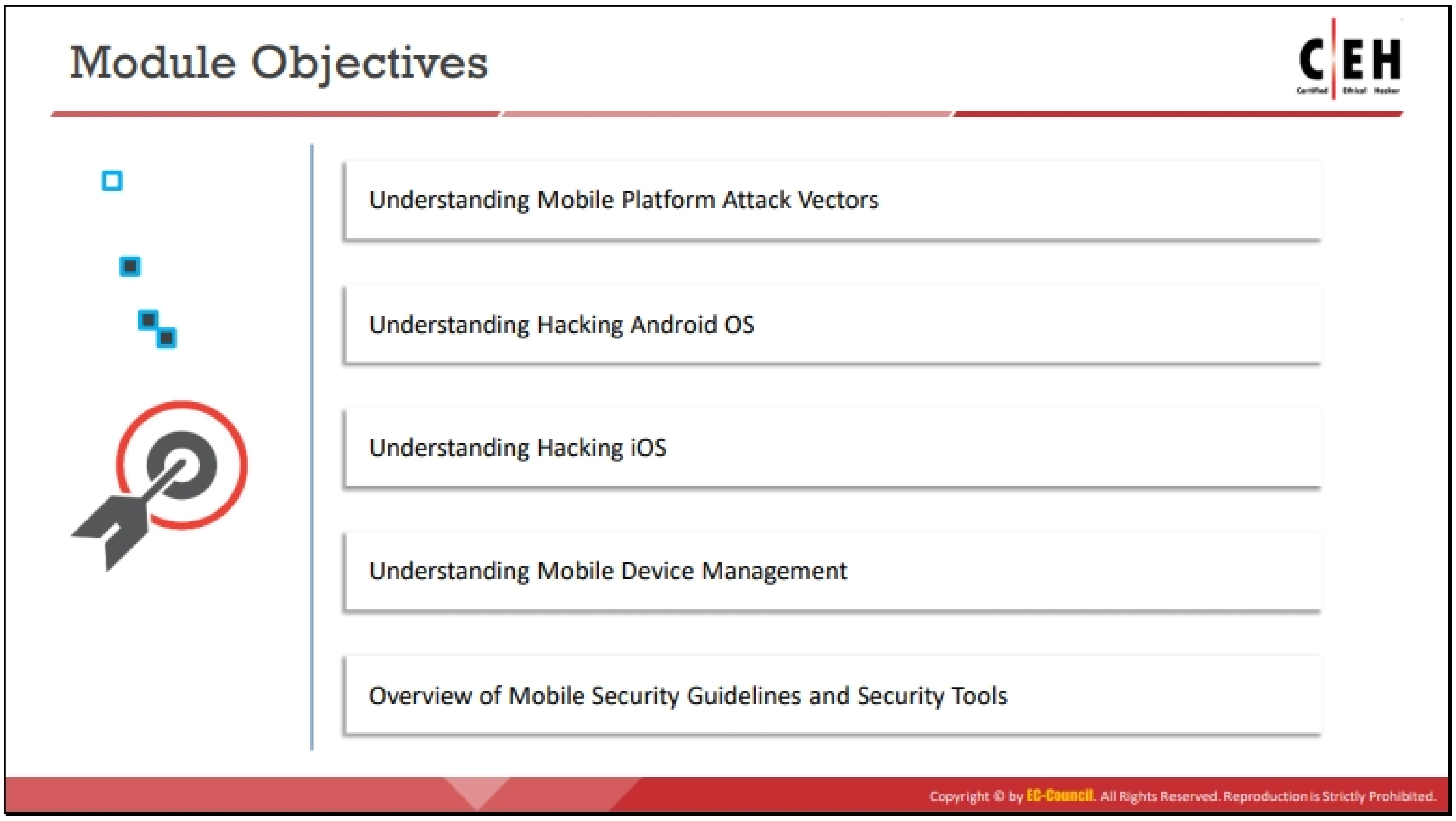
C|EH™

Certified Ethical Hacker

Module 17:

Hacking Mobile Platforms



Module Objectives

- Understanding Mobile Platform Attack Vectors
- Understanding Hacking Android OS
- Understanding Hacking iOS
- Understanding Mobile Device Management
- Overview of Mobile Security Guidelines and Security Tools

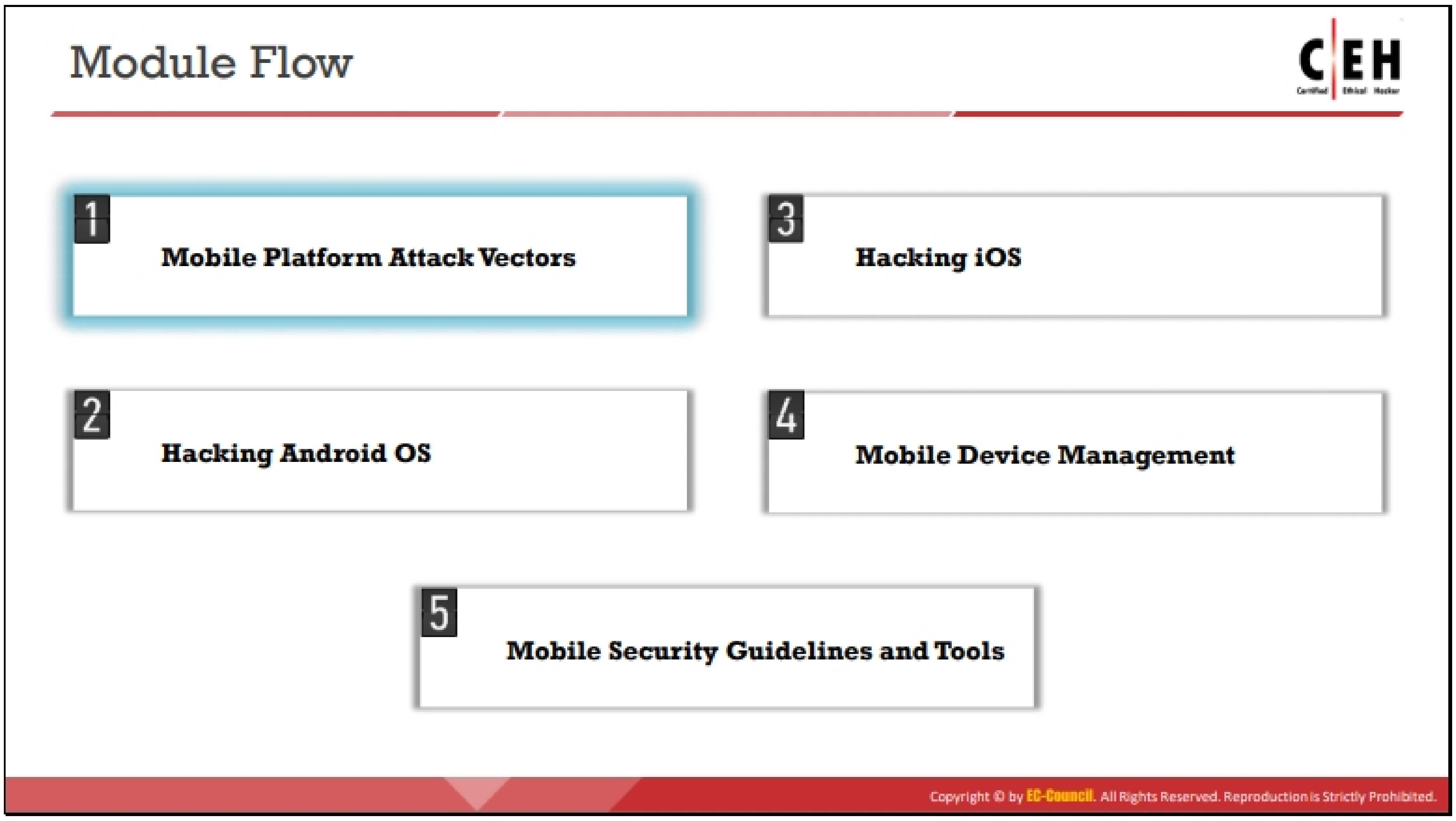
Module Objectives

With the advancement of mobile technology, mobility has become the key parameter for Internet usage. People's lifestyles are becoming increasingly reliant on smartphones and tablets. Mobile devices are replacing desktops and laptops, as they allow users to not only access the Internet, email, and GPS navigation but also store critical data such as contact lists, passwords, calendars, and login credentials. In addition, recent developments in mobile commerce have enabled users to perform transactions such as purchasing goods and applications over wireless networks, redeeming coupons and tickets, and banking from their smartphones.

Believing that surfing the Internet on mobile devices is safe, many users fail to enable existing security software. The popularity of smartphones and their moderately strong security mechanisms have made them attractive targets for attackers. This module explains the potential threats to mobile platforms and provides guidelines for using mobile devices securely.

At the end of this module, you will be able to:

- Understand mobile platform attack vectors
- Explain how to hack Android OS
- Explain how to hack iOS
- Understand the importance of mobile device management (MDM)
- Adopt various mobile security countermeasures
- Use various mobile security tools



Module Flow

1 **Mobile Platform Attack Vectors**

2 **Hacking Android OS**

3 **Hacking iOS**

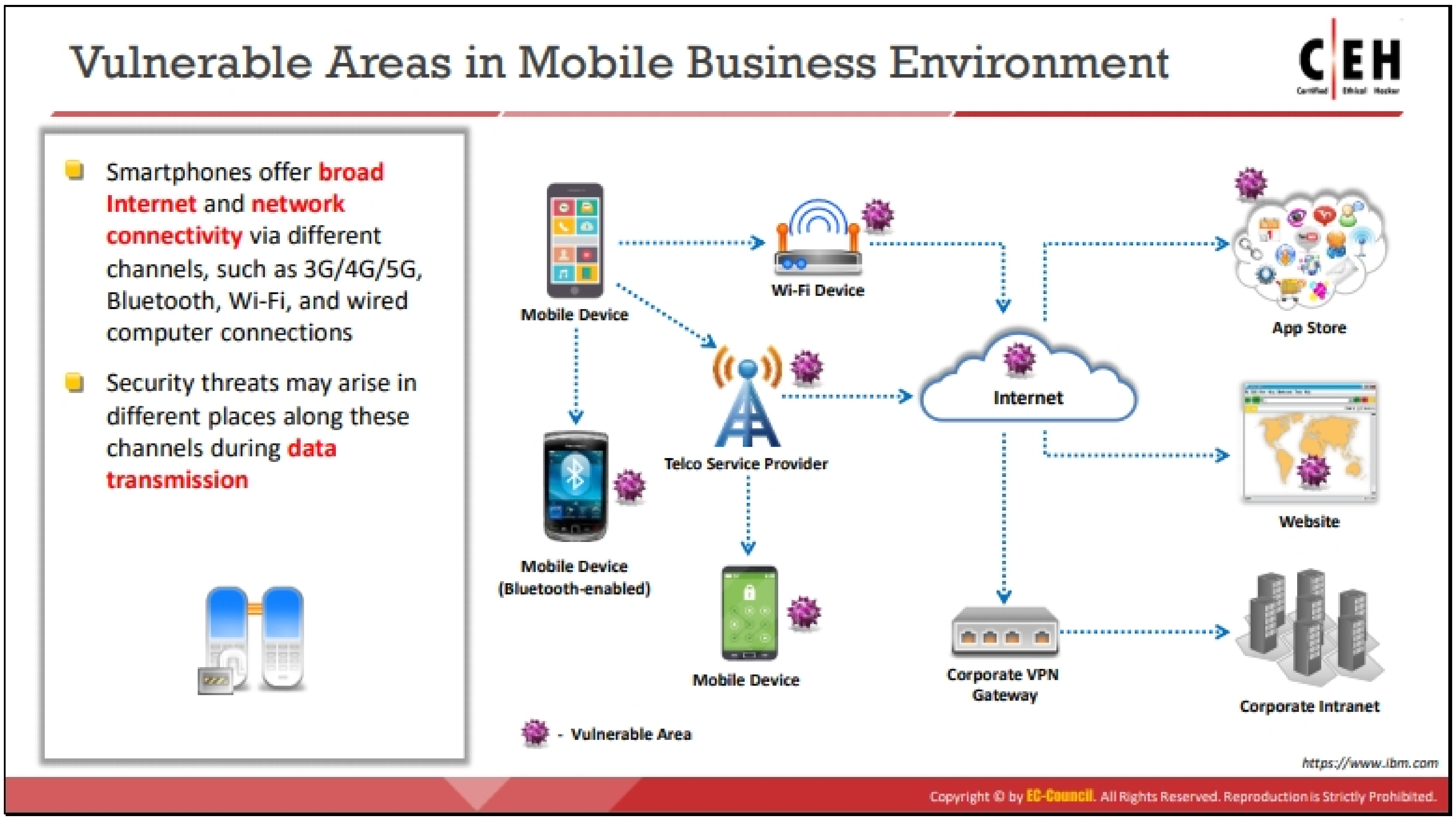
4 **Mobile Device Management**

5 **Mobile Security Guidelines and Tools**

Mobile Platform Attack Vectors

Mobile security is becoming increasingly challenging with the emergence of complex attacks that use multiple attack vectors to compromise mobile devices. These security threats exploit critical data as well as financial information and other details of mobile users and may also damage the reputation of mobile networks and organizations.

This section discusses vulnerable areas in the mobile business environment, the OWASP top 10 mobile risks, the anatomy of mobile attacks, mobile attack vectors, associated vulnerabilities and risks, security issues arising from app stores, app sandboxing issues, mobile spam, pairing mobile devices on open Bluetooth and Wi-Fi connections, and other mobile attacks.



Vulnerable Areas in Mobile Business Environment

- Smartphones offer **broad Internet** and **network connectivity** via different channels, such as 3G/4G/5G, Bluetooth, Wi-Fi, and wired computer connections
- Security threats may arise in different places along these channels during **data transmission**

C|EH
Mobile Device
Wi-Fi Device
App Store
Telco Service Provider
Internet
Website
Mobile Device (Bluetooth-enabled)
Mobile Device
Corporate VPN Gateway
Corporate Intranet
- Vulnerable Area

https://www.ibm.com

Vulnerable Areas in Mobile Business Environment

Source: *https://www.ibm.com*

Smartphones are being widely used for both business and personal purposes. Thus, they are a treasure trove for attackers who seek to steal corporate or personal data. Security threats to mobile devices have increased because of the increase in Internet connectivity, the use of commercial and other applications, different methods of communication, and so on. Apart from the security threats that are specific to mobile devices, mobile devices are also susceptible to many other threats that are applicable to desktop and laptop computers, web applications, networks, etc.

Nowadays, smartphones offer Internet and network connectivity via various channels such as 3G/4G/5G, Bluetooth, Wi-Fi, or a wired computer connection. Security threats may arise at different places along these paths during data transmission.

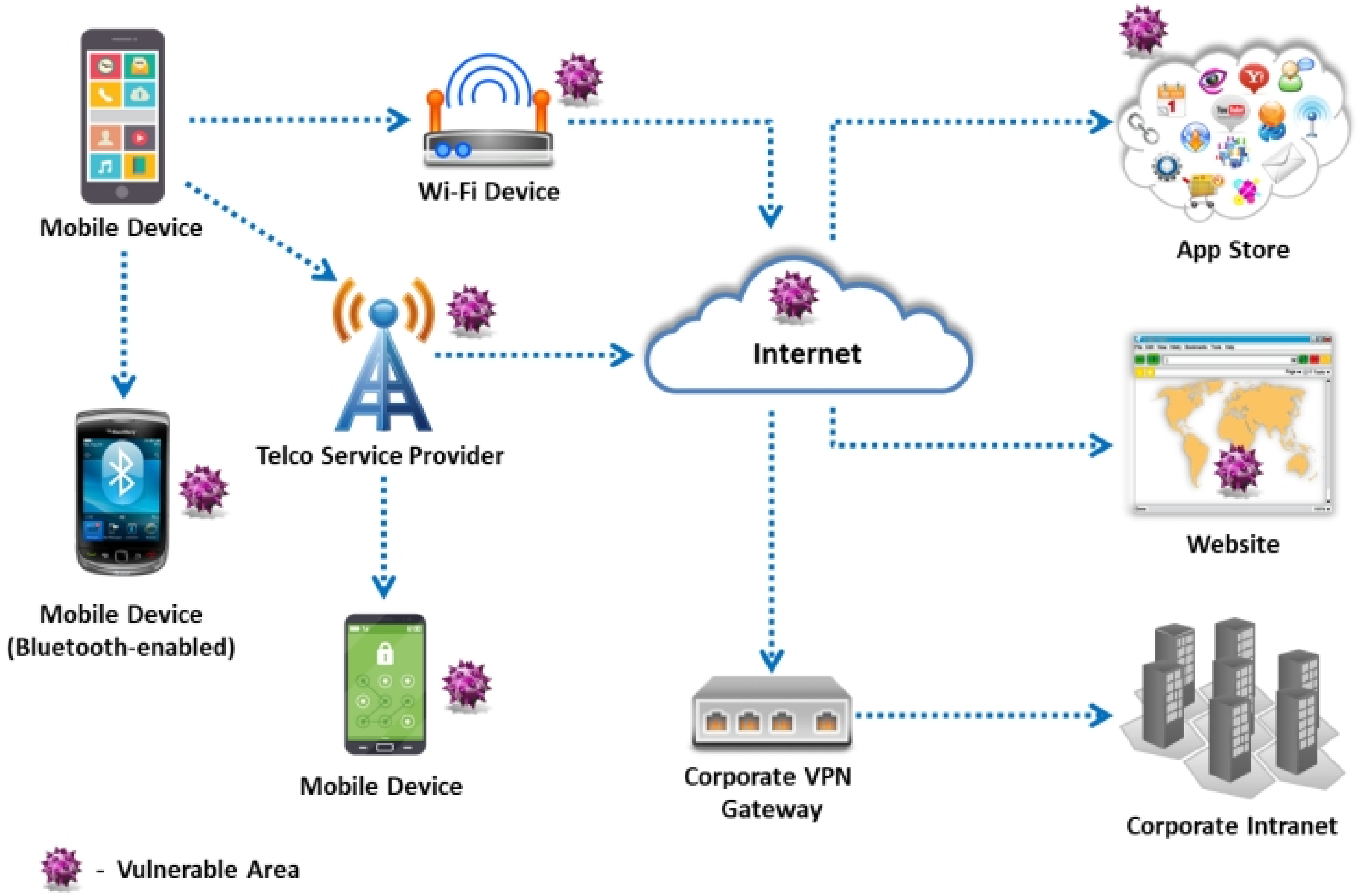
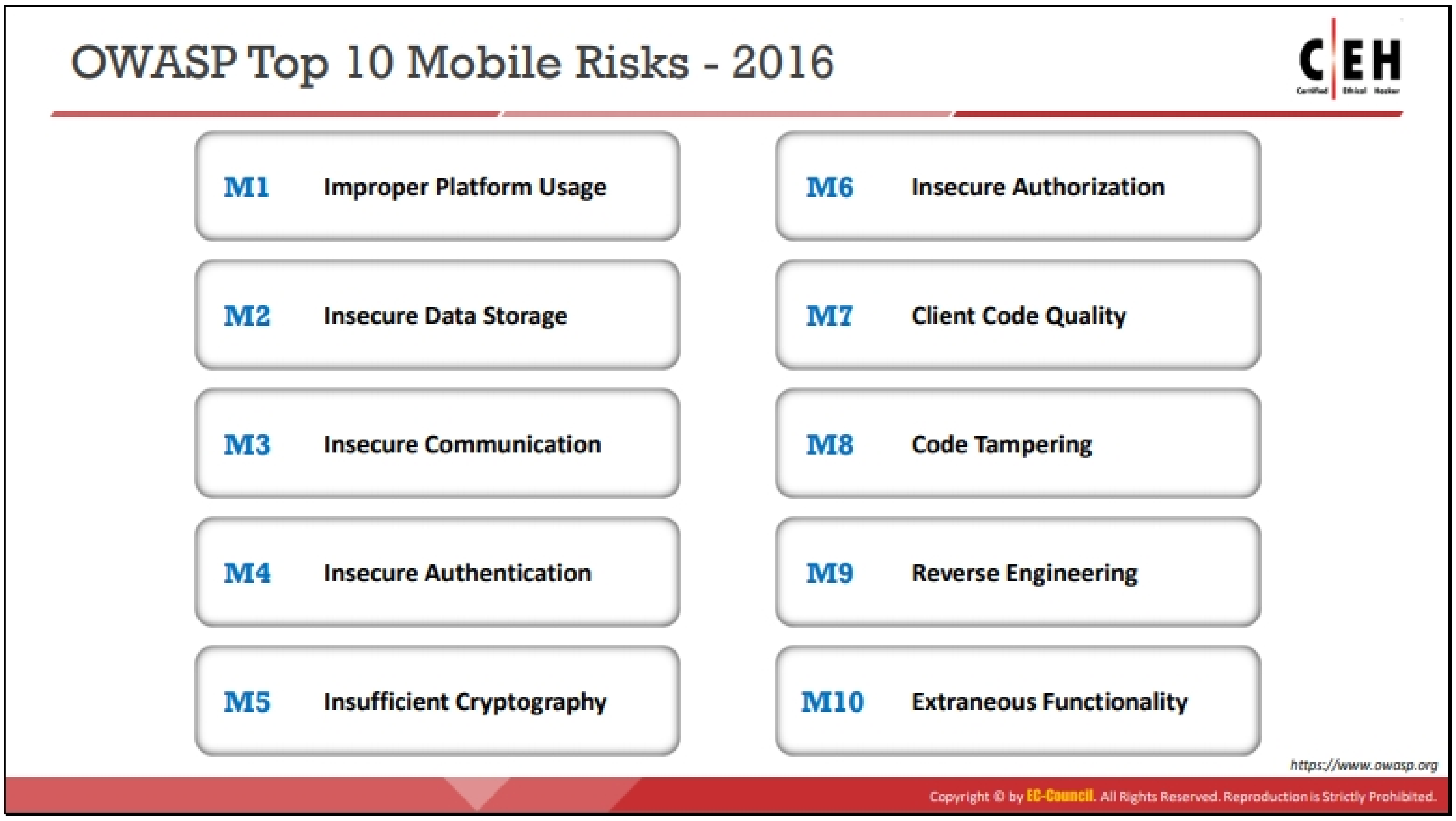

Mobile Device
Wi-Fi Device
App Store
Internet
Telco Service Provider
Mobile Device
(Bluetooth-enabled)
Website
Mobile Device
Corporate VPN
Gateway
Corporate Intranet
- Vulnerable Area

Figure 17.1: Vulnerable areas in the mobile business environment



OWASP Top 10 Mobile Risks - 2016

M1	**Improper Platform Usage**
M2	**Insecure Data Storage**
M3	**Insecure Communication**
M4	**Insecure Authentication**
M5	**Insufficient Cryptography**
M6	**Insecure Authorization**
M7	**Client Code Quality**
M8	**Code Tampering**
M9	**Reverse Engineering**
M10	**Extraneous Functionality**

https://www.owasp.org

OWASP Top 10 Mobile Risks - 2016

Source: *https://www.owasp.org*

According to OWASP, the following are the top 10 mobile risks:

- **M1—Improper Platform Usage**

  This category covers the misuse of a platform feature or the failure to use platform security controls. It includes Android intents, platform permissions, and the misuse of Touch ID, Keychain, or some other security control that is part of the mobile device's OS. There are several ways in which mobile apps can be exposed to this risk.

- **M2—Insecure Data Storage**

  Insecure data storage vulnerability arises when development teams assume that users and malware will not have access to a mobile device's file system and subsequently to sensitive information in the device's data stores. "Jailbreaking" or rooting a mobile device bypasses encryption protection mechanisms. OWASP recommends analyzing platforms' data security application programming interfaces (APIs) and calling them appropriately.

  Unintended data leakage occurs when a developer unintentionally places sensitive data in a location on the mobile device that is easily accessible by other apps on the device. Such leakage is normally caused by vulnerabilities in the OS, frameworks, compiler environment, new hardware, and so on without a developer's knowledge. It is a significant threat to the OS, platforms, and frameworks; thus, it is important to understand how they handle features such as URL caching, browser cookie objects, and HTML5 data storage.

- **M3—Insecure Communication**

  This category covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, and so on. Such flaws expose an individual user's data and can lead to account theft. If the adversary intercepts an admin account, the entire site could be exposed. A poor Secure Socket Layer (SSL) setup can also facilitate phishing and man-in-the-middle (MITM) attacks.

- **M4—Insecure Authentication**

  This category captures notions of authenticating the end user or bad session management such as

  - Failing to identify the user when it is required
  - Failure to maintain the user's identity when it is required.
  - Weaknesses in session management.

- **M5—Insufficient Cryptography**

  The code applies cryptography to a sensitive information asset. However, cryptography is insufficient in some ways. This category covers issues in which cryptography is attempted but not performed correctly. This vulnerability will result in the unauthorized retrieval of sensitive information from the mobile device. To exploit this weakness, an adversary must successfully convert encrypted code or sensitive data into its original unencrypted form due to weak encryption algorithms or flaws in the process of encryption.

- **M6—Insecure Authorization**

  This category captures failures in authorization (e.g., authorization decisions on the client side and forced browsing). It is distinct from authentication issues (e.g., device enrolment and user identification).

  When an app does not authenticate users at all in a situation where it should (e.g., granting anonymous access to some resource or service when authenticated and authorized access is required), then it is an authentication failure and not an authorization failure.

- **M7—Client Code Quality**

  This category covers "Security Decisions via Untrusted Inputs" and is one of the less frequently used categories. It is the catch-all for code-level implementation problems in the mobile client, which are distinct from server-side coding mistakes. It captures buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that is running on the mobile device. Most exploitations that fall into this category result in foreign code execution or DoS on remote server endpoints (and not the mobile device itself).

- **M8—Code Tampering**

  This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification.

  Once an application is delivered to a mobile device, its code and data resources are resident on the device. An attacker can directly modify the code, change the memory contents dynamically, change or replace the system APIs that the application uses, or modify the application's data and resources. Thus, the attacker can directly subvert the intended use of the software for personal or monetary gain.

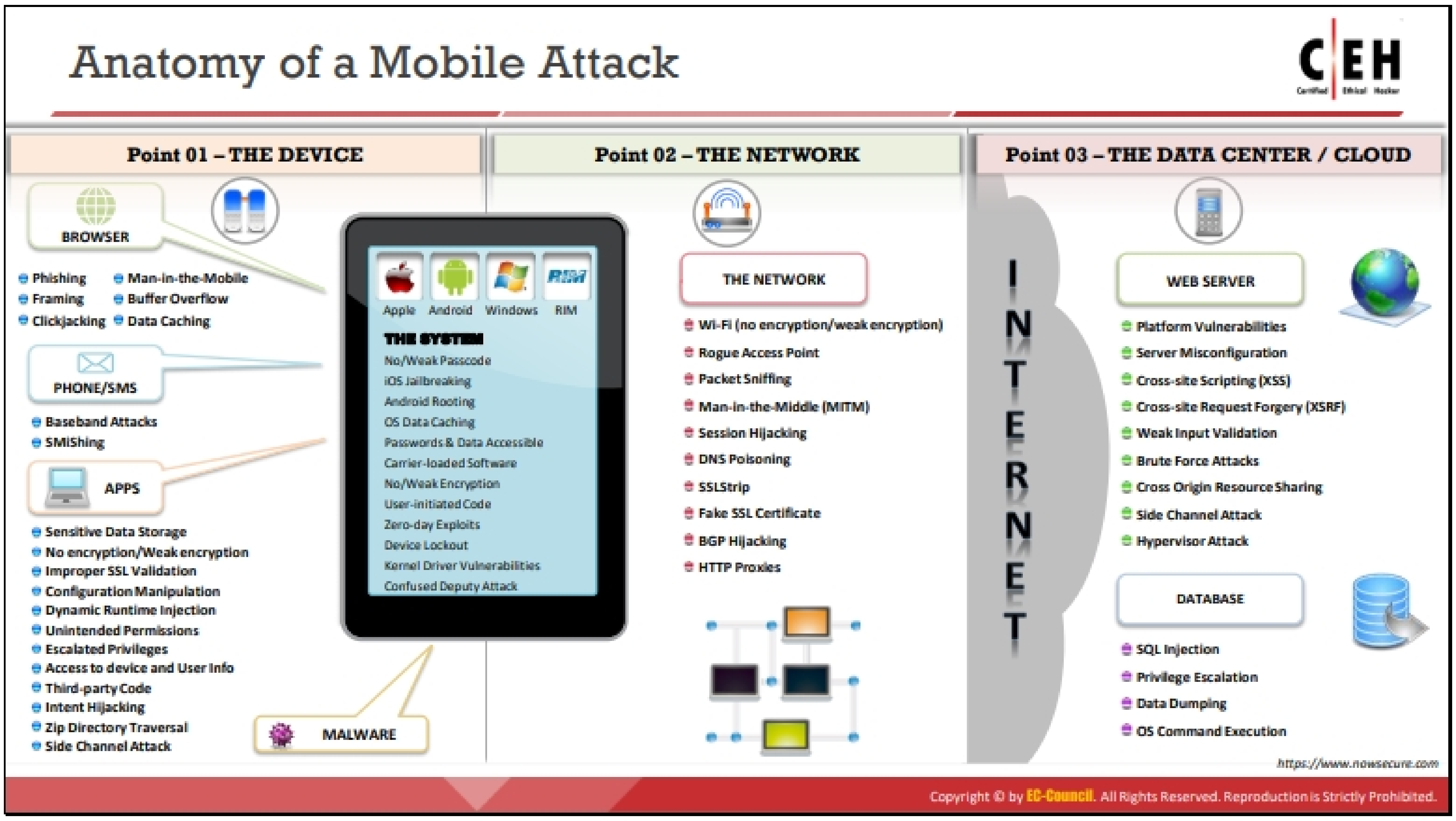
- **M9—Reverse Engineering**

  This category includes the analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. Software such as IDA, Hopper, otool, and other binary inspection tools give the attacker insights into the inner workings of the application. Thus, he/she may exploit other nascent vulnerabilities in the application and uncover information about backend servers, cryptographic constants and ciphers, and intellectual property.

- **M10—Extraneous Functionality**

  Often, developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a hybrid app. Another example involves the disabling of two-factor authentication during testing.

  Typically, an attacker seeks to understand extraneous functionality within a mobile app to discover hidden functionality in the backend systems. Attackers will typically exploit such extraneous functionality directly from their own systems without any involvement by the end users.



Anatomy of a Mobile Attack

Point 01 – THE DEVICE

BROWSER

- Phishing
- Framing
- Clickjacking
- Man-in-the-Mobile
- Buffer Overflow
- Data Caching

PHONE/SMS

- Baseband Attacks
- SMiShing

APPS

- Sensitive Data Storage
- No encryption/Weak encryption
- Improper SSL Validation
- Configuration Manipulation
- Dynamic Runtime Injection
- Unintended Permissions
- Escalated Privileges
- Access to device and User Info
- Third-party Code
- Intent Hijacking
- Zip Directory Traversal
- Side Channel Attack

Apple Android Windows RIM

THE SYSTEM

- No/Weak Passcode
- iOS Jailbreaking
- Android Rooting
- OS Data Caching
- Passwords & Data Accessible
- Carrier-loaded Software
- No/Weak Encryption
- User-initiated Code
- Zero-day Exploits
- Device Lockout
- Kernel Driver Vulnerabilities
- Confused Deputy Attack

MALWARE

Point 02 – THE NETWORK

THE NETWORK

- Wi-Fi (no encryption/weak encryption)
- Rogue Access Point
- Packet Sniffing
- Man-in-the-Middle (MITM)
- Session Hijacking
- DNS Poisoning
- SSLStrip
- Fake SSL Certificate
- BGP Hijacking
- HTTP Proxies

INTERNET

Point 03 – THE DATA CENTER / CLOUD

WEB SERVER

- Platform Vulnerabilities
- Server Misconfiguration
- Cross-site Scripting (XSS)
- Cross-site Request Forgery (XSRF)
- Weak Input Validation
- Brute Force Attacks
- Cross Origin Resource Sharing
- Side Channel Attack
- Hypervisor Attack

DATABASE

- SQL Injection
- Privilege Escalation
- Data Dumping
- OS Command Execution

https://www.nowsecure.com

Anatomy of a Mobile Attack

Source: *https://www.nowsecure.com*

Because of the extensive usage and implementation of bring your own device (BYOD) policies in organizations, mobile devices have emerged as a prime target for attacks. Attackers scan these devices for vulnerabilities. Such attacks can involve the device and the network layer, the data center, or a combination of them.

Attackers exploit vulnerabilities associated with the following to launch malicious attacks:

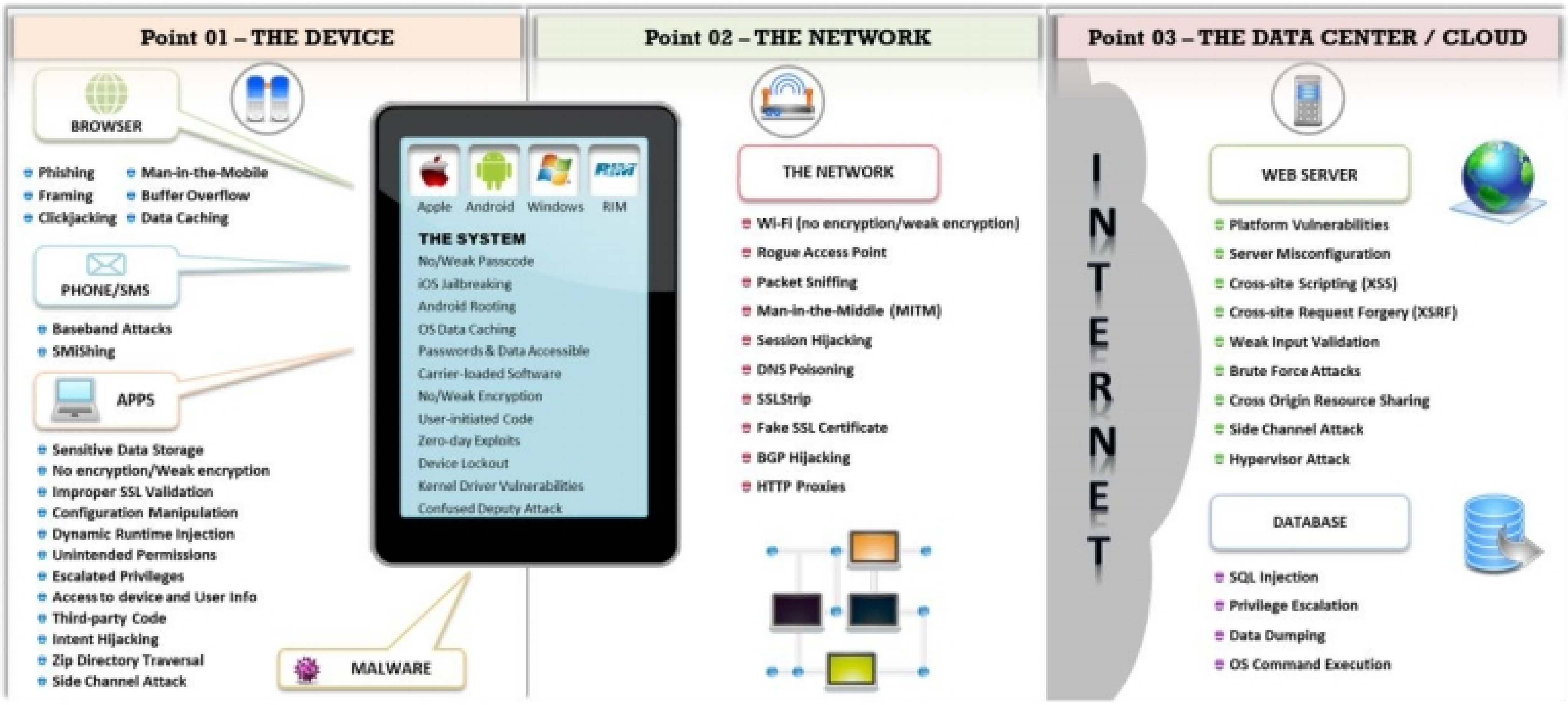

Point 01 – THE DEVICE
BROWSER
Phishing
Framing
Clickjacking
Man-in-the-Mobile
Buffer Overflow
Data Caching
PHONE/SMS
Baseband Attacks
SMiShing
APPS
Sensitive Data Storage
No encryption/Weak encryption
Improper SSL Validation
Configuration Manipulation
Dynamic Runtime Injection
Unintended Permissions
Escalated Privileges
Access to device and User Info
Third-party Code
Intent Hijacking
Zip Directory Traversal
Side Channel Attack
Apple
Android
Windows
RIM
THE SYSTEM
No/Weak Passcode
iOS Jailbreaking
Android Rooting
OS Data Caching
Passwords & Data Accessible
Carrier-loaded Software
No/Weak Encryption
User-initiated Code
Zero-day Exploits
Device Lockout
Kernel Driver Vulnerabilities
Confused Deputy Attack
MALWARE
Point 02 – THE NETWORK
THE NETWORK
Wi-Fi (no encryption/weak encryption)
Rogue Access Point
Packet Sniffing
Man-in-the-Middle (MITM)
Session Hijacking
DNS Poisoning
SSLStrip
Fake SSL Certificate
BGP Hijacking
HTTP Proxies
INTERNET
Point 03 – THE DATA CENTER / CLOUD
WEB SERVER
Platform Vulnerabilities
Server Misconfiguration
Cross-site Scripting (XSS)
Cross-site Request Forgery (XSRF)
Weak Input Validation
Brute Force Attacks
Cross Origin Resource Sharing
Side Channel Attack
Hypervisor Attack
DATABASE
SQL Injection
Privilege Escalation
Data Dumping
OS Command Execution

Figure 17.2: Anatomy of a mobile attack

- **The Device**

  Vulnerabilities in mobile devices pose significant risks to sensitive personal and corporate data. Attackers targeting the device itself can use various entry points.

  Device-based attacks are of the following types:

  - **Browser-based Attacks**

    Browser-based methods of attack are as follows:

    - **Phishing**: Phishing emails or pop-ups redirect users to fake web pages that mimic trustworthy sites, asking them to submit their personal information such as username, password, credit card details, address, and mobile number. Mobile users are more likely to be victims of phishing sites because the devices are small in size and they display only short URLs, limited warning messages, scaled-down lock icons, and so on.
    - **Framing**: Framing involves a web page integrated into another web page using the iFrame elements of HTML. An attacker exploits iFrame functionality used in the target website, embeds his/her malicious web page, and uses clickjacking to steal users' sensitive information.
    - **Clickjacking**: Clickjacking, also known as a user interface redress attack, is a malicious technique used to trick web users into clicking something different from what they think they are clicking. Consequently, attackers obtain sensitive information or take control of the device.
    - **Man-in-the-Mobile**: An attacker implants malicious code into the victim's mobile device to bypass password verification systems that send one-time passwords (OTPs) via SMS or voice calls. Thereafter, the malware relays the gathered information to the attacker.
    - **Buffer Overflow**: Buffer overflow is an abnormality whereby a program, while writing data to a buffer, surfeits the intended limit and overwrites the adjacent memory. This results in erratic program behavior, including memory access errors, incorrect results, and mobile device crashes.
    - **Data Caching**: Data caches in mobile devices store information that is often required by these devices to interact with web applications, thereby preserving scarce resources and resulting in better responses time for client applications. Attackers attempt to exploit these data caches to access the sensitive information stored in them.

  - **Phone/SMS-based Attacks**

    Phone/SMS-based methods of attack are as follows:

    - **Baseband Attacks**: Attackers exploit vulnerabilities in a phone's GSM/3GPP baseband processor, which sends and receives radio signals to cell towers.

- **SMiShing**: SMS phishing (also known as SMiShing) is a type of phishing fraud in which an attacker uses SMS to send text messages containing deceptive links of malicious websites or telephone numbers to a victim. The attacker tricks the victim into clicking the link or calling the phone number and revealing his or her personal information such as social security number (SSN), credit card number, and online banking username and password.

- **Application-based Attacks**

  Application-based methods of attack are as follows:

  - **Sensitive Data Storage**: Some apps installed and used by mobile users employ weak security in their database architecture, which makes them targets for attackers who seek to hack and steal the sensitive user information stored in them.
  - **No Encryption/Weak Encryption**: Apps that transmit unencrypted or weakly encrypted data are susceptible to attacks such as session hijacking.
  - **Improper SSL Validation**: Security loopholes in an application's SSL validation process may allow attackers to circumvent the data security.
  - **Configuration Manipulation**: Apps may use external configuration files and libraries that can be exploited in a configuration manipulation attack. This includes gaining unauthorized access to administration interfaces and configuration stores as well as retrieval of clear text configuration data.
  - **Dynamic Runtime Injection**: Attackers manipulate and abuse the run time of an application to circumvent security locks and logic checks, access privileged parts of an app, and even steal data stored in memory.
  - **Unintended Permissions**: Misconfigured apps can sometimes open doors to attackers by providing unintended permissions.
  - **Escalated Privileges**: Attackers engage in privilege escalation attacks, which take advantage of design flaws, programming errors, bugs, or configuration oversights to gain access to resources that are usually protected from an application or user.

  Other application-based methods of attack include UI overlay/pin stealing, third-party code, intent hijacking, zip directory traversal, clipboard data, URL schemes, GPS spoofing, weak/no local authentication, integrity/tampering/repackaging, side-channel attack, app signing key unprotected, app transport security, XML specialization, and so on.

- **The System**

  OS-based methods of attack are as follows:

- **No Passcode/Weak Passcode**: Many users choose not to set a passcode or use a weak PIN, passcode, or pattern lock, which an attacker can easily guess or crack to compromise sensitive data stored in the mobile device.
- **iOS Jailbreaking**: Jailbreaking iOS is the process of removing the security mechanisms set by Apple to prevent malicious code from running on the device. It provides root access to the OS and removes sandbox restrictions. Thus, jailbreaking involves many security risks as well as other risks to iOS devices, including poor performance, malware infection, and so on.
- **Android Rooting**: Rooting allows Android users to attain privileged control (known as "root access") within Android's subsystem. Like jailbreaking, rooting can result in the exposure of sensitive data stored in the mobile device.
- **OS Data Caching**: An OS cache stores used data/information in memory on a temporary basis in the hard disk. An attacker can dump this memory by rebooting the victim's device with a malicious OS and extract sensitive data from the dumped memory.
- **Passwords and Data Accessible**: iOS devices store encrypted passwords and data using cryptographic algorithms that have certain known vulnerabilities. Attackers exploit these vulnerabilities to decrypt the device's Keychain, exposing user passwords, encryption keys, and other private data.
- **Carrier-loaded Software**: Pre-installed software or apps on devices may contain vulnerabilities that an attacker can exploit to perform malicious activities such as deleting, modifying, or stealing data on the device, eavesdropping on calls, and so on.
- **User-initiated Code**: User-initiated code is an activity that tricks the victim into installing malicious applications or clicking links that allow an attacker to install malicious code to exploit the user's browser, cookies, and security permissions.

Other OS-based methods of attack include no/weak encryption, confused deputy attack, TEE/secure enclave processor, side-channel leakage, multimedia/file format parsers, kernel driver vulnerabilities, resource DoS, GPS spoofing, device lockout, and so on.

- **The Network**

  Network-based methods of attack are as follows:

  - **Wi-Fi (weak encryption/no encryption)**: Some applications fail to encrypt data or use weak algorithms to encrypt data for transmission across wireless networks. An attacker may intercept the data by eavesdropping on the wireless connection. Although many applications use SSL/TLS, which offers protection for data in transit, attacks against these algorithms can expose users' sensitive information.

  - **Rogue Access Points**: Attackers install an illicit wireless access point by physical means, which allows them to access a protected network by hijacking the connections of legitimate network users.
  - **Packet Sniffing**: An attacker uses sniffing tools such as Wireshark and Capsa Network Analyzer to capture and analyze all the data packets in network traffic, which generally include sensitive data such as login credentials sent in clear text.
  - **Man-in-the-Middle (MITM)**: Attackers eavesdrop on existing network connections between two systems, intrude into these connections, and then read or modify the data or insert fraudulent data into the intercepted communication.
  - **Session Hijacking**: Attackers steal valid session IDs and use them to gain unauthorized access to user and network information.
  - **DNS Poisoning**: Attackers exploit network DNS servers, resulting in the substitution of false IP addresses at the DNS level. Thus, website users are directed to another website of the attacker's choice.
  - **SSLStrip**: SSLStrip is a type of MITM attack in which attackers exploit vulnerabilities in the SSL/TLS implementation on websites. It relies on the user validating the presence of the HTTPS connection. The attack invisibly downgrades connections to HTTP without encryption, which is difficult for users to detect in mobile browsers.
  - **Fake SSL Certificates**: Fake SSL certificates represent another type of MITM attack in which an attacker issues a fake SSL certificate to intercept traffic on a supposedly secure HTTPS connection.

    Other network-based methods of attack include BGP hijacking, HTTP proxies, etc.

- **The Data Center/CLOUD**

  Data centers have two primary points of entry: a web server and a database.

  - **Web-server-based attacks**

    Web-server-based vulnerabilities and attacks are of the following types:

    - **Platform Vulnerabilities**: Attackers exploit vulnerabilities in the OS, server software such as IIS, or application modules running on the web server. Sometimes, attackers can expose vulnerabilities associated with the protocol or access controls by monitoring the communication established between a mobile device and a web server.
    - **Server Misconfiguration**: A misconfigured web server may allow an attacker to gain unauthorized access to its resources.
    - **Cross-site Scripting (XSS)**: XSS attacks exploit vulnerabilities in dynamically generated web pages, which enable malicious attackers to inject client-side script into web pages viewed by other users. Such attacks occur when invalidated input data are included in dynamic content sent to the user's web browser for rendering. Attackers inject malicious JavaScript, VBScript, ActiveX,

HTML, or Flash for execution on a victim's system by hiding it within legitimate requests.

- **Cross-Site Request Forgery (CSRF)**: CSRF attacks exploit web page vulnerabilities that allow an attacker to force an unsuspecting user's browser to send unintended malicious requests. The victim holds an active session with a trusted site and simultaneously visits a malicious site that injects an HTTP request for the trusted site into his/her session, compromising its integrity.
- **Weak Input Validation**: Web services excessively trust the input from mobile applications, depending on the application to perform input validation. However, attackers can forge their own communication to the web server or circumvent the app's logic checks, allowing them to take advantage of missing validation logic on the server to perform unauthorized actions.

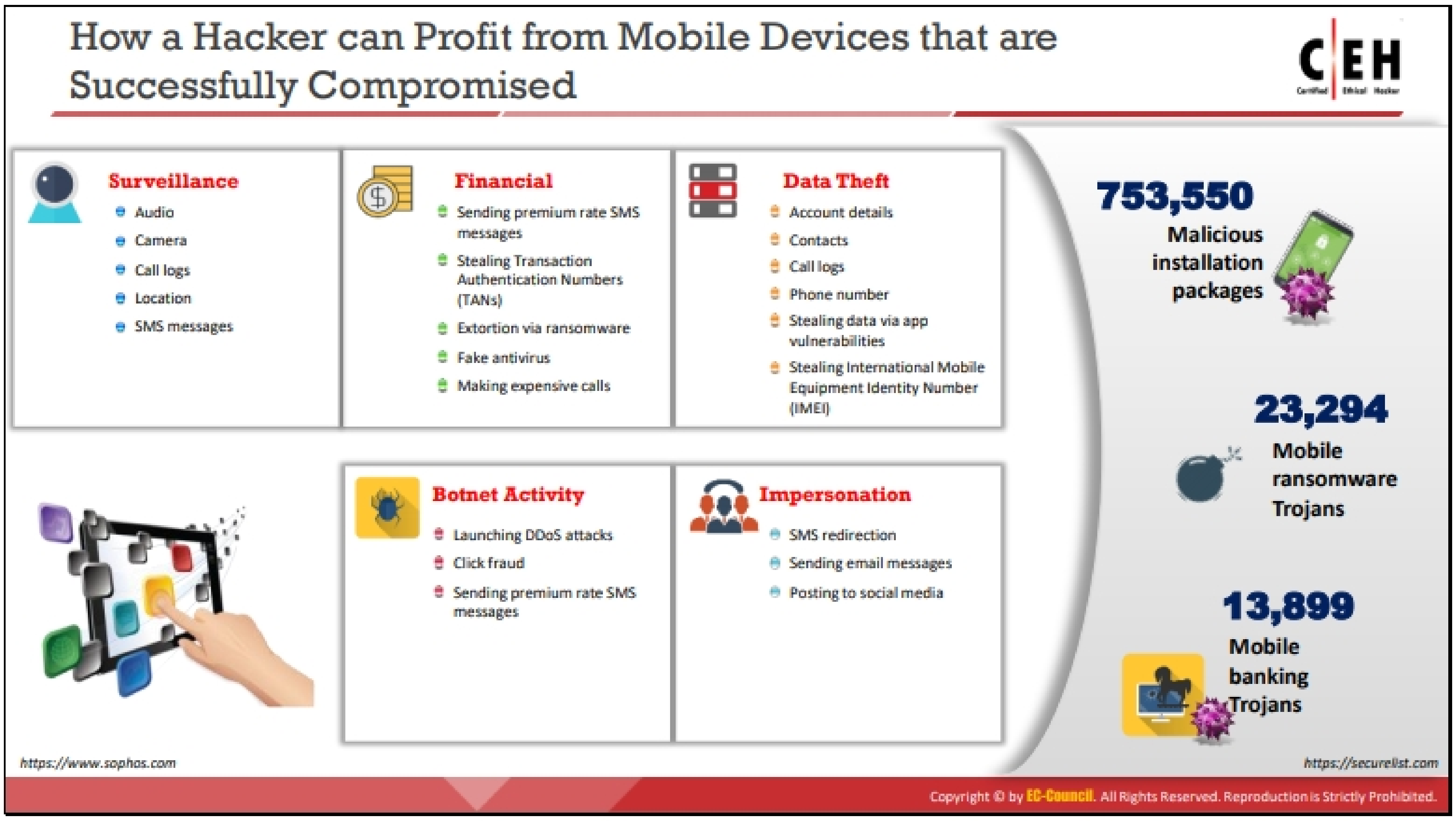
  Attackers exploit input validation flaws so that they can perform cross-site scripting, buffer overflow, injection attacks, and so on, which lead to data theft and system malfunction.
- **Brute-Force Attacks**: Attackers adopt the trial-and-error approach to guess the valid input to a particular field. Applications that allow any number of input attempts are generally prone to brute-force attacks.

  Other web-server-based vulnerabilities and attacks include cross-origin resource sharing, side-channel attack, hypervisor attack, VPN, and so on.

- **Database Attacks**

  Database-based vulnerabilities and attacks are of the following types:

  - **SQL injection**: SQL injection is a technique used to take advantage of nonvalidated input vulnerabilities to pass SQL commands through a web application for execution by a backend database. It is a basic attack used to gain unauthorized access to a database or to retrieve information directly from the database.
  - **Privilege Escalation**: This occurs when an attack leverages some exploit to gain high-level access, resulting in the theft of sensitive data stored in the database.
  - **Data Dumping**: An attacker causes the database to dump some or all of its data, thereby uncovering sensitive records.
  - **OS Command Execution**: An attacker injects OS-level commands into a query, causing certain database systems to execute these commands on the server. Thus, the attacker can gain unrestricted/root-level system access.



How a Hacker can Profit from Mobile Devices that are Successfully Compromised

Surveillance
- Audio
- Camera
- Call logs
- Location
- SMS messages

Financial
- Sending premium rate SMS messages
- Stealing Transaction Authentication Numbers (TANs)
- Extortion via ransomware
- Fake antivirus
- Making expensive calls

Data Theft
- Account details
- Contacts
- Call logs
- Phone number
- Stealing data via app vulnerabilities
- Stealing International Mobile Equipment Identity Number (IMEI)

Botnet Activity
- Launching DDoS attacks
- Click fraud
- Sending premium rate SMS messages

Impersonation
- SMS redirection
- Sending email messages
- Posting to social media

753,550 Malicious installation packages

23,294 Mobile ransomware Trojans

13,899 Mobile banking Trojans

https://www.sophos.com

https://securelist.com

How a Hacker can Profit from Mobile Devices that are Successfully Compromised

Source: *https://www.sophos.com*, *https://securelist.com*

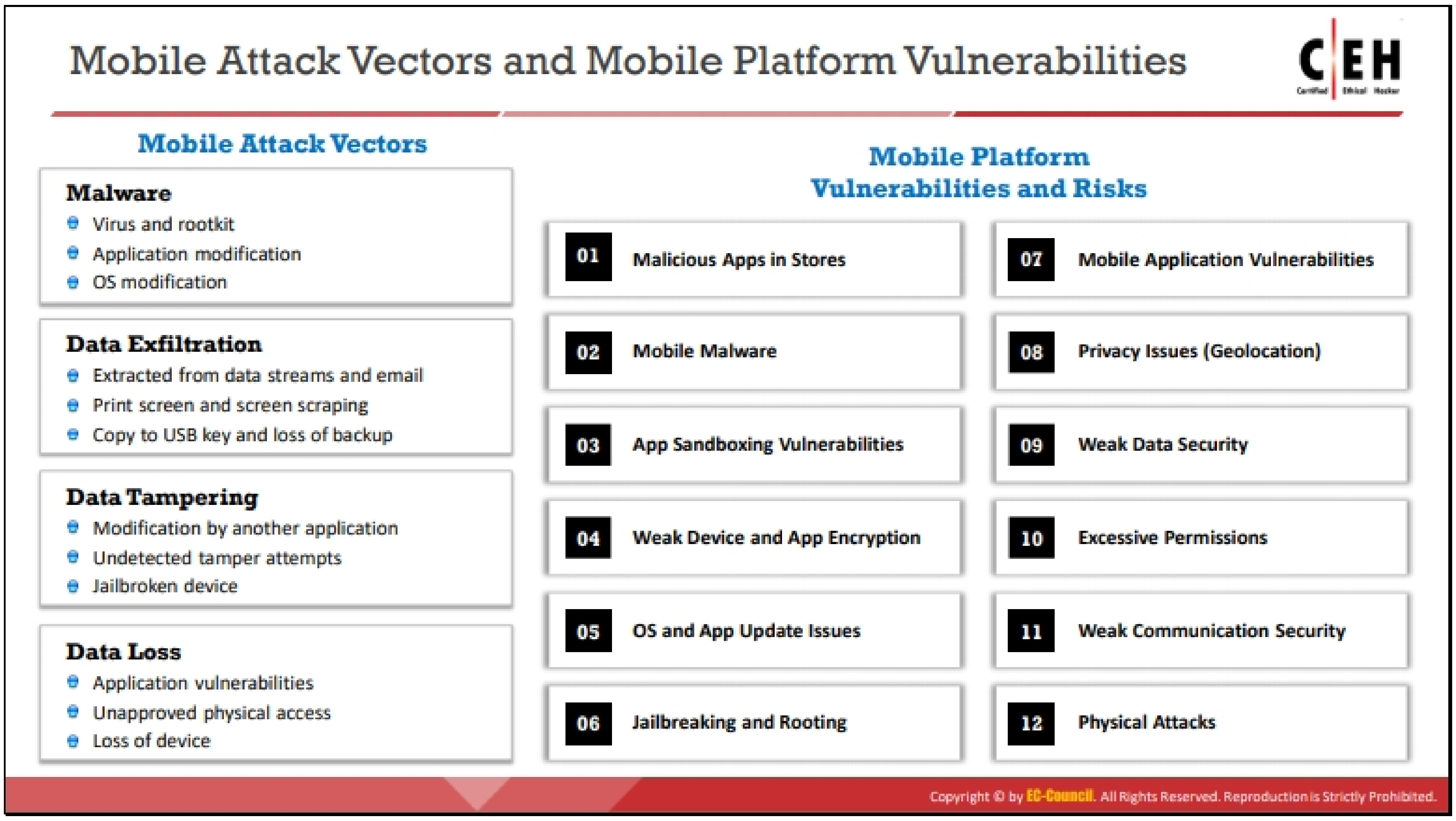
At present, images, contact lists, banking apps, social media apps, email accounts, financial information, business information, and so on reside on our smartphone devices. Thus, smartphones are a treasure trove of information for potential exploitation by attackers. Android devices are particularly likely to be hacked, as they account for the majority of the mobile market share.

Upon compromising a smartphone, an attacker can spy on user activities, misuse the sensitive information stolen, impersonate the user by posting on his/her social media accounts, or enlist the device in a botnet (a network of many hacked smartphones).

After successfully compromising the mobile device, hackers can exploit the following:

Surveillance	**Financial**	**Data Theft**	**Botnet Activity**	**Impersonation**
Audio	Sending premium-rate SMS messages	Account details	Launching DDoS attacks	SMS redirection
Camera	Fake anti-virus	Contacts	Click fraud	Sending emails
Call logs	Making expensive calls	Call logs and phone number	Sending premium-rate SMS messages	Posting to social media
Location	Extortion via ransomware	Stealing data via app vulnerabilities		
SMS messages	Stealing Transaction Authentication Numbers (TANs)	Stealing International Mobile Equipment Identity Number (IMEI)		

Table 17.1: List of information that hackers can exploit



Mobile Attack Vectors and Mobile Platform Vulnerabilities

Mobile Attack Vectors

Malware
- Virus and rootkit
- Application modification
- OS modification

Data Exfiltration
- Extracted from data streams and email
- Print screen and screen scraping
- Copy to USB key and loss of backup

Data Tampering
- Modification by another application
- Undetected tamper attempts
- Jailbroken device

Data Loss
- Application vulnerabilities
- Unapproved physical access
- Loss of device

Mobile Platform Vulnerabilities and Risks

01 Malicious Apps in Stores
02 Mobile Malware
03 App Sandboxing Vulnerabilities
04 Weak Device and App Encryption
05 OS and App Update Issues
06 Jailbreaking and Rooting
07 Mobile Application Vulnerabilities
08 Privacy Issues (Geolocation)
09 Weak Data Security
10 Excessive Permissions
11 Weak Communication Security
12 Physical Attacks

Mobile Attack Vectors and Mobile Platform Vulnerabilities

- **Mobile Attack Vectors**

  Mobile devices have attracted the attention of attackers owing to their widespread use. Such devices access many of the resources that traditional computers use. Moreover, these devices have some unique features that have led to the emergence of new attack vectors and protocols. Such vectors make mobile phone platforms susceptible to malicious attacks both from the network and upon physical compromise. Given below are some of the attack vectors that allow an attacker to exploit vulnerabilities in mobile OS, device firmware, or mobile apps.

Malware	Data Exfiltration	Data Tampering	Data Loss
Virus and rootkit	Extracted from data streams and email	Modification by another application	Application vulnerabilities
Application modification	Print screen and screen scraping	Undetected tamper attempts	Unapproved physical access
OS modification	Copy to USB key and loss of backup	Jailbroken device	Loss of device

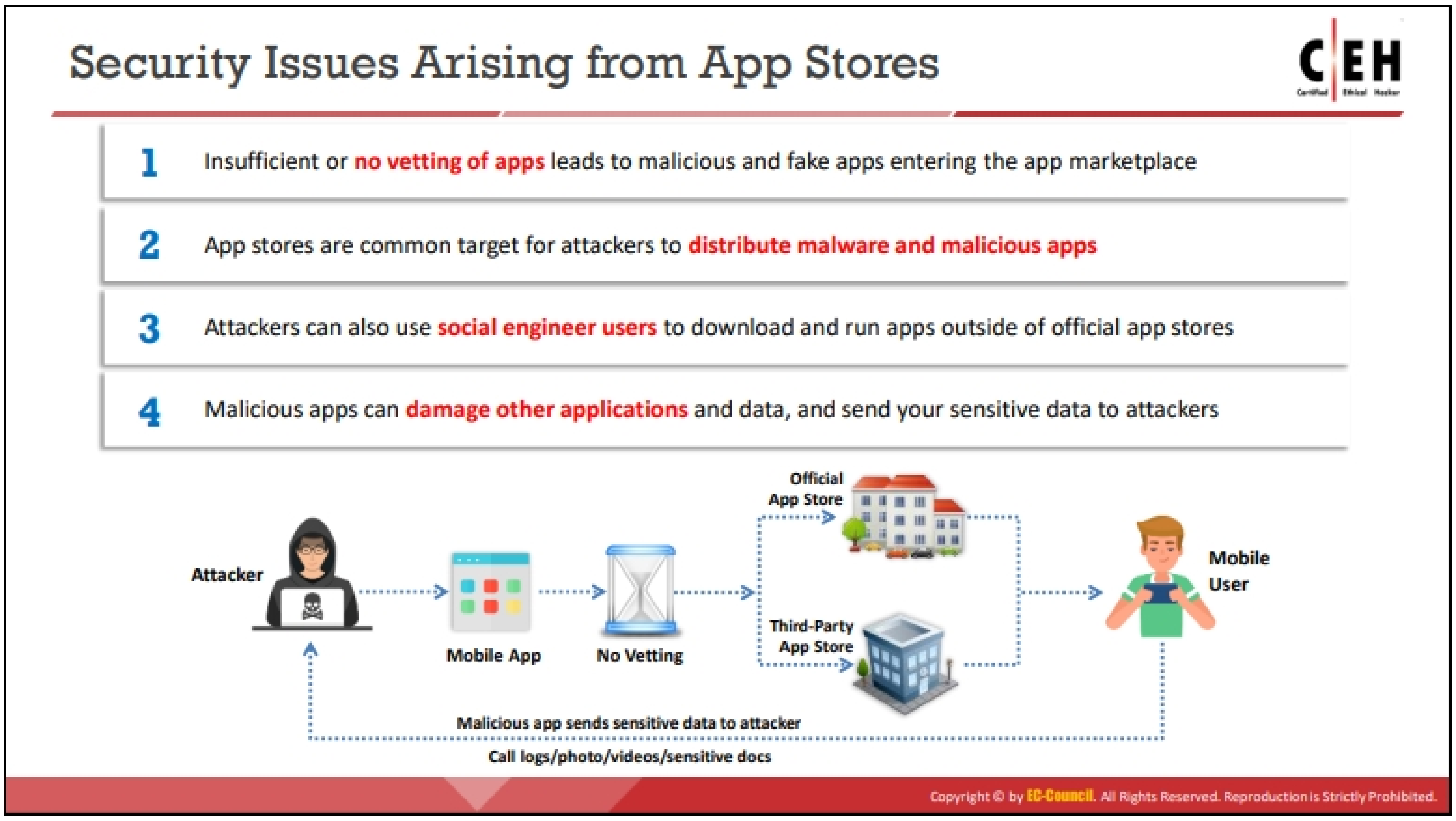
Table 17.2: List of attack vectors

- **Mobile Platform Vulnerabilities and Risks**

  The growing use of smartphones with ever-evolving technological features has made mobile device security a primary security concern for the IT sector. Mobile devices are becoming privileged targets for cyber criminals because of significant improvements in both mobile OS and hardware. In addition, the enhancements in smartphone features introduce new types of security concerns. As smartphones are surpassing PCs as preferred devices to access the Internet, manage communications, and so on, attackers are more attracted toward mobile research and implement possible attack schemes against mobile platforms to compromise users' security and privacy or even gain complete control over the victims' devices.

Some mobile platform vulnerabilities and risks are listed below:

- Malicious apps in stores
- Mobile malware
- App sandboxing vulnerabilities
- Weak device and app encryption
- OS and app update issues
- Jailbreaking and rooting
- Mobile application vulnerabilities
- Privacy issues (Geolocation)
- Weak data security
- Excessive permissions
- Weak communication security
- Physical attacks
- Insufficient code obfuscation
- Insufficient transport layer security
- Insufficient session expiration



Security Issues Arising from App Stores

1 Insufficient or **no vetting of apps** leads to malicious and fake apps entering the app marketplace

2 App stores are common target for attackers to **distribute malware and malicious apps**

3 Attackers can also use **social engineer users** to download and run apps outside of official app stores

4 Malicious apps can **damage other applications** and data, and send your sensitive data to attackers

Attacker
Mobile App
No Vetting
Official App Store
Third-Party App Store
Mobile User
Malicious app sends sensitive data to attacker
Call logs/photo/videos/sensitive docs

Security Issues Arising from App Stores

Mobile applications are computer programs designed to run on smartphones, tablets, and other mobile devices. Such applications include text messaging, email, playing videos and music, voice recording, games, banking, shopping, and so on. In general, apps are made available via application distribution platforms, which could be official app stores operated by the owners of mobile OS, such as Apple's App Store, Google Play app store, and Microsoft App Store, or third-party app stores such as Amazon Appstore, GetJar, and APKMirror.

App stores are common targets for attackers who seek to distribute malware and malicious apps. Attackers may download a legitimate app, repackage it with malware, and upload it to a third-party app store, from which users download it, considering it to be genuine. Malicious apps installed on user systems can damage other applications or stored data and send sensitive data such as call logs, photos, videos, sensitive docs, and so on to the attacker without the users' knowledge. Attackers may use the information gathered to exploit the devices and launch further attacks. Attackers can also perform social engineering, which forces users to download and run apps outside the official app stores. Insufficient or no vetting of apps usually leads to the entry of malicious and fake apps in the marketplace. Malicious apps can damage other applications and data and send users' sensitive data to attackers.

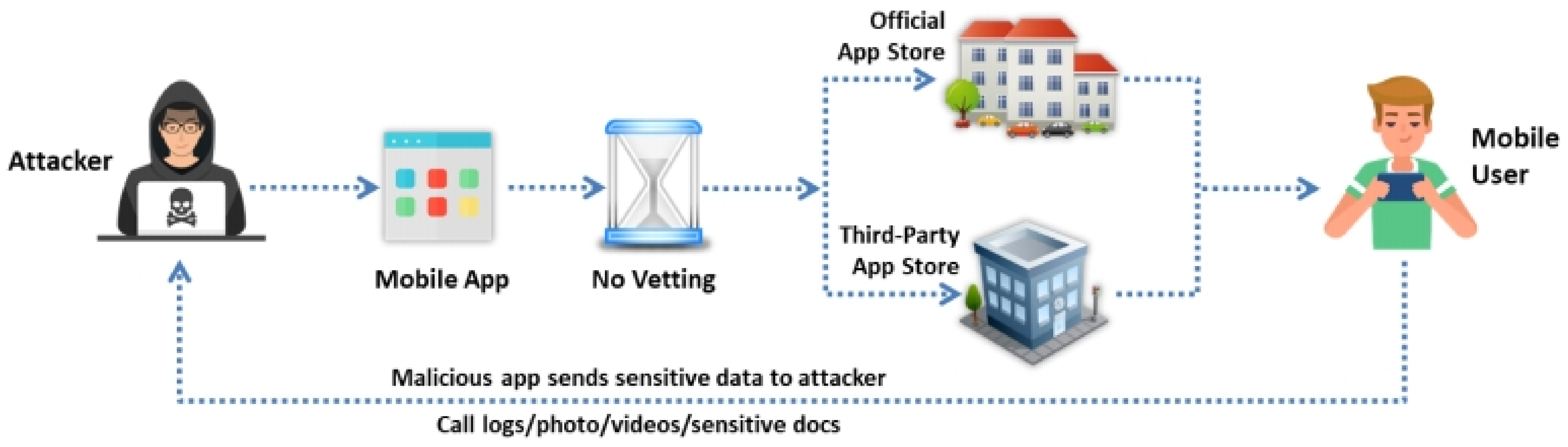
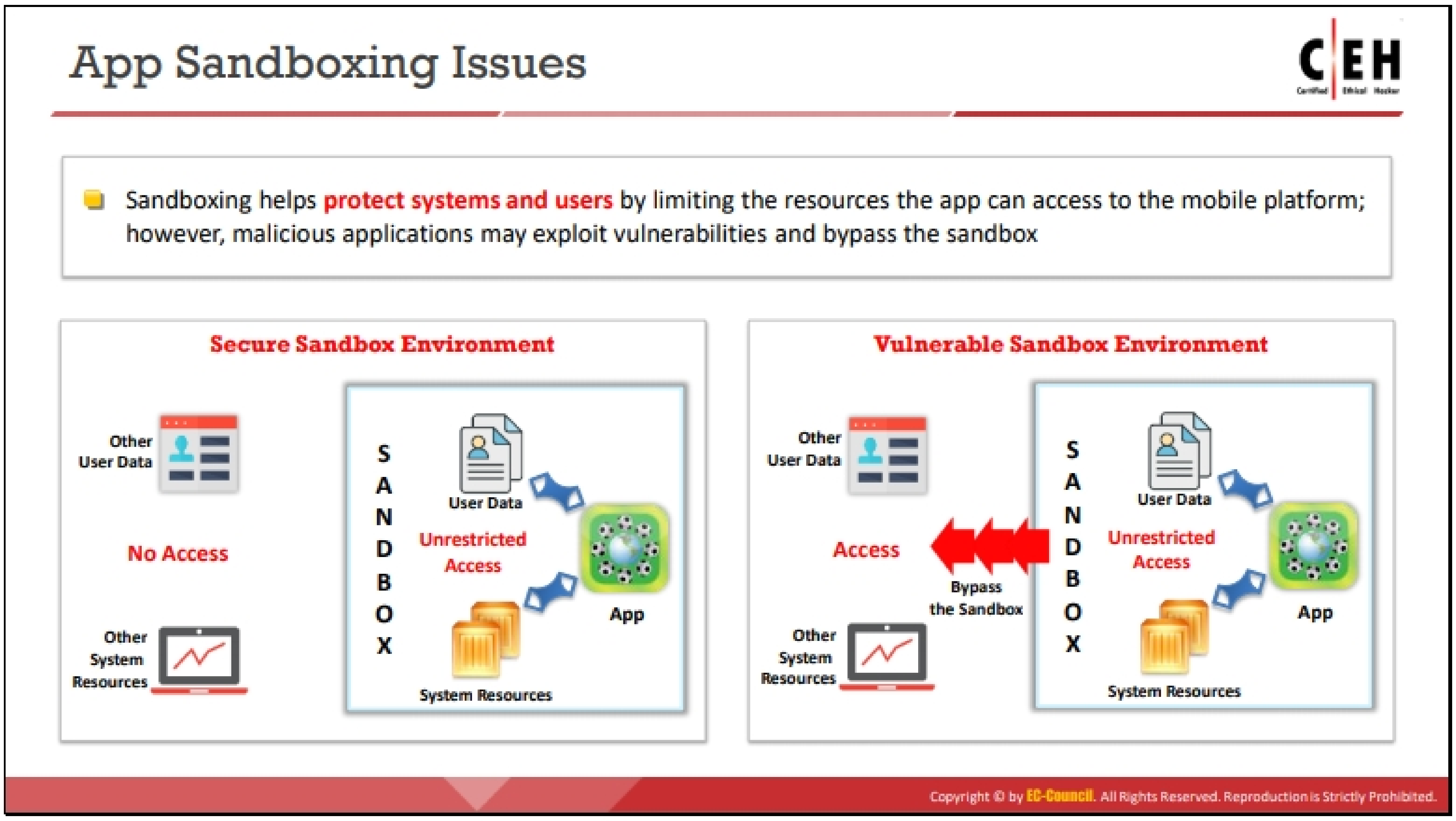

Attacker
Mobile App
No Vetting
Official App Store
Third-Party App Store
Mobile User
Malicious app sends sensitive data to attacker
Call logs/photo/videos/sensitive docs

Figure 17.3: Security Issues Arising from App Stores



App Sandboxing Issues

- Sandboxing helps **protect systems and users** by limiting the resources the app can access to the mobile platform; however, malicious applications may exploit vulnerabilities and bypass the sandbox

Secure Sandbox Environment
Other User Data
No Access
Other System Resources
SANDBOX
User Data
Unrestricted Access
App
System Resources
Vulnerable Sandbox Environment
Other User Data
Access
Bypass the Sandbox
Other System Resources
SANDBOX
User Data
Unrestricted Access
App
System Resources

App Sandboxing Issues

Smartphones are increasingly attracting the attention of cyber criminals. Mobile app developers must understand the threat to the security and privacy of mobile devices by running a non-sandboxed app, and they should develop sandboxed apps accordingly.

App sandboxing is a security mechanism that helps protect systems and users by limiting the resources that an app can access to its intended functionality on the mobile platform. Often, sandboxing is useful in executing untested code or untrusted programs from unverified or untrusted third parties, suppliers, users, and websites. This enhances security by isolating the app to prevent intruders, system resources, malware such as Trojans and viruses, and other apps from interacting with it. As sandboxing isolates applications from one another, it protects them from tampering with each other; however, malicious applications may exploit vulnerabilities and bypass the sandbox.

A secure sandbox environment provides an application with limited privileges intended for its functionality to restrict it from accessing other users' data and system resources, whereas a vulnerable sandbox environment allows a malicious application to exploit vulnerabilities in the sandbox and breach its perimeter, resulting in the exploitation of other data and system resources.

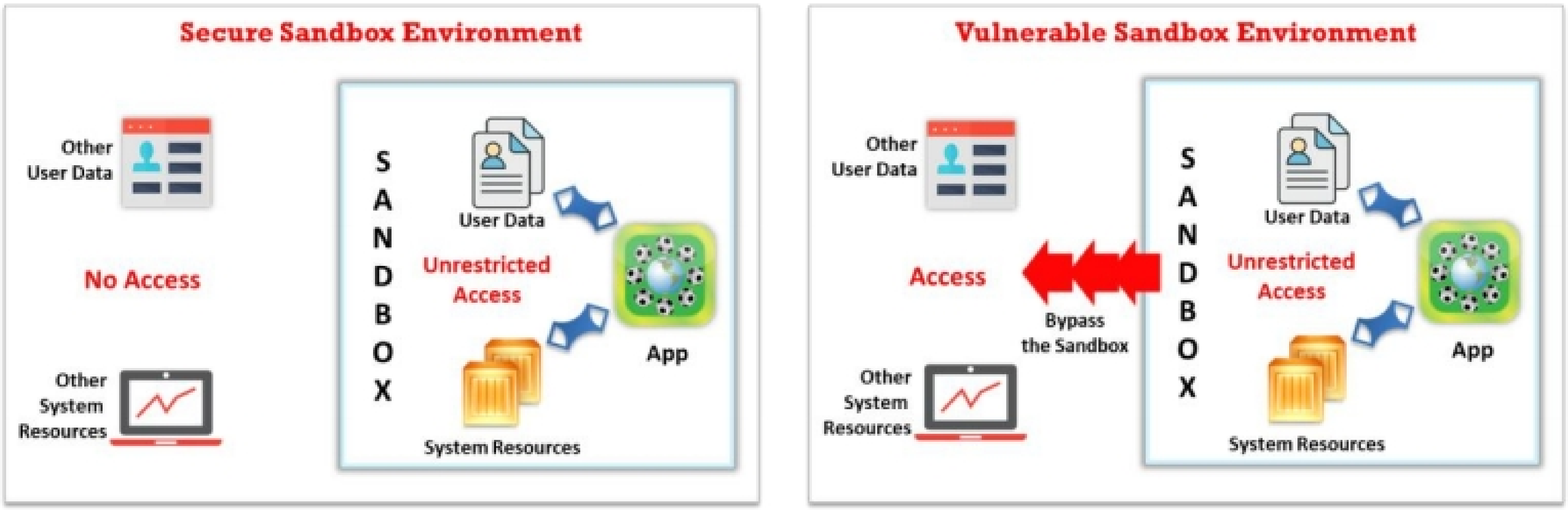
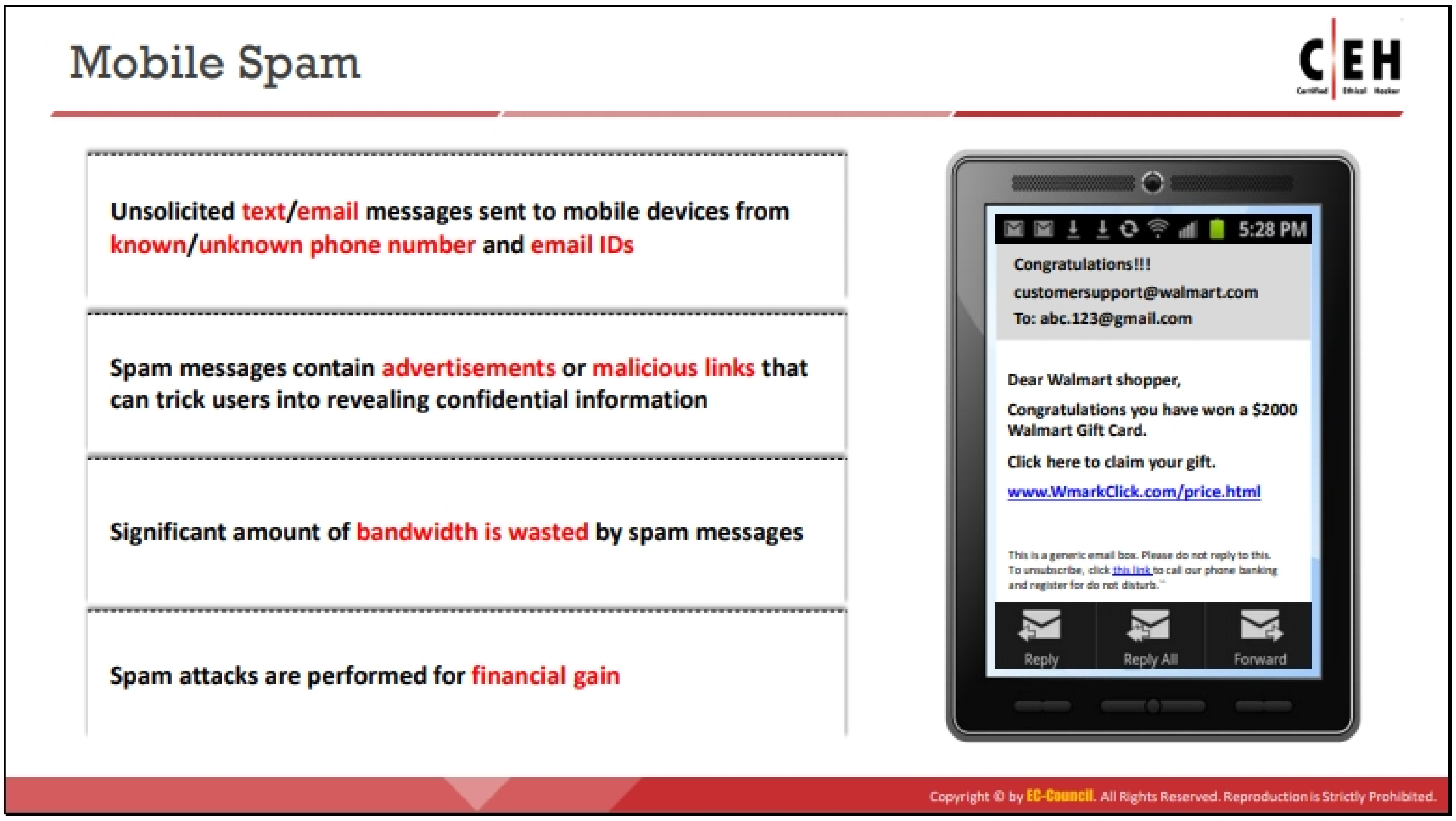

Secure Sandbox Environment
Other User Data
No Access
Other System Resources
SANDBOX
User Data
Unrestricted Access
App
System Resources
Vulnerable Sandbox Environment
Other User Data
Access
Bypass the Sandbox
Other System Resources
SANDBOX
User Data
Unrestricted Access
App
System Resources

Figure 17.4: App Sandboxing issues



Mobile Spam

Unsolicited text/email messages sent to mobile devices from known/unknown phone number and email IDs

Spam messages contain advertisements or malicious links that can trick users into revealing confidential information

Significant amount of bandwidth is wasted by spam messages

Spam attacks are performed for financial gain

Congratulations!!!
customersupport@walmart.com
To: abc.123@gmail.com

Dear Walmart shopper,

Congratulations you have won a $2000 Walmart Gift Card.

Click here to claim your gift.

www.WmarkClick.com/price.html

This is a generic email box. Please do not reply to this. To unsubscribe, click this link to call our phone banking and register for do not disturb."

Mobile Spam

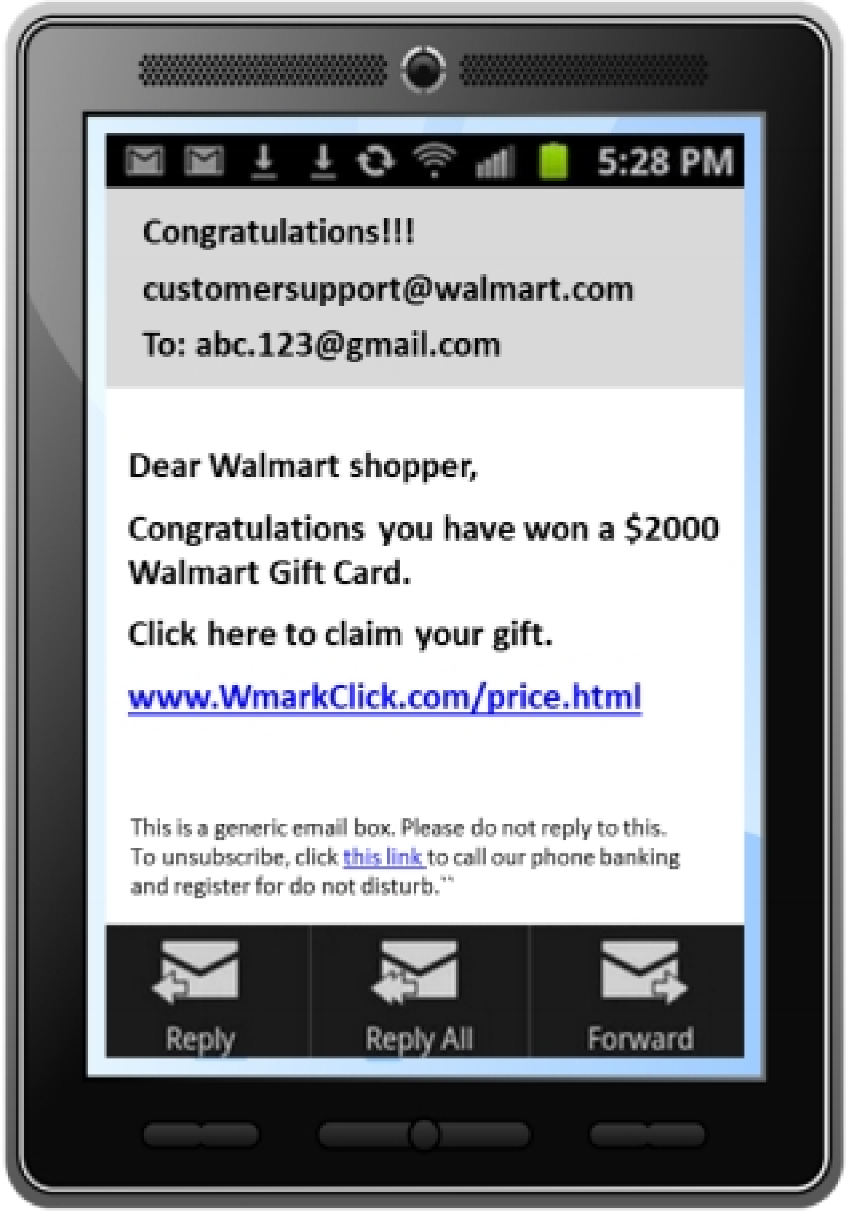
At present, mobile phones are widely used for both personal and business purposes. Spam is a generic term for unsolicited messages sent via electronic communication technologies such as SMS, MMS, instant messaging (IM), and email.

Mobile phone spam, also known as SMS spam, text spam, or m-spam, refers to unsolicited messages sent in bulk form to known/unknown phone numbers/email IDs to target mobile phones.

Typical spam messages delivered to mobile phones are as follows:

- Messages containing advertisements or malicious links that can trick users into revealing confidential information
- Attractive commercial messages advertising products/services
- SMS or MMS messages claiming that the victim has won a prize and asking him/her to place a call to a provided premium-rate telephone service number for further details
- Malicious links that may lure users into divulging sensitive personal or corporate data
- Phishing messages that lure the recipient into revealing personal or financial data such as name, address, date of birth, bank account number, credit card number, and so on, which an attacker can use to commit identity or financial fraud

Spam messages consume a significant amount of network bandwidth. The consequences of mobile spam include financial loss, malware injection, and corporate data breach incidents.



Congratulations!!!
customersupport@walmart.com
To: abc.123@gmail.com

Dear Walmart shopper,

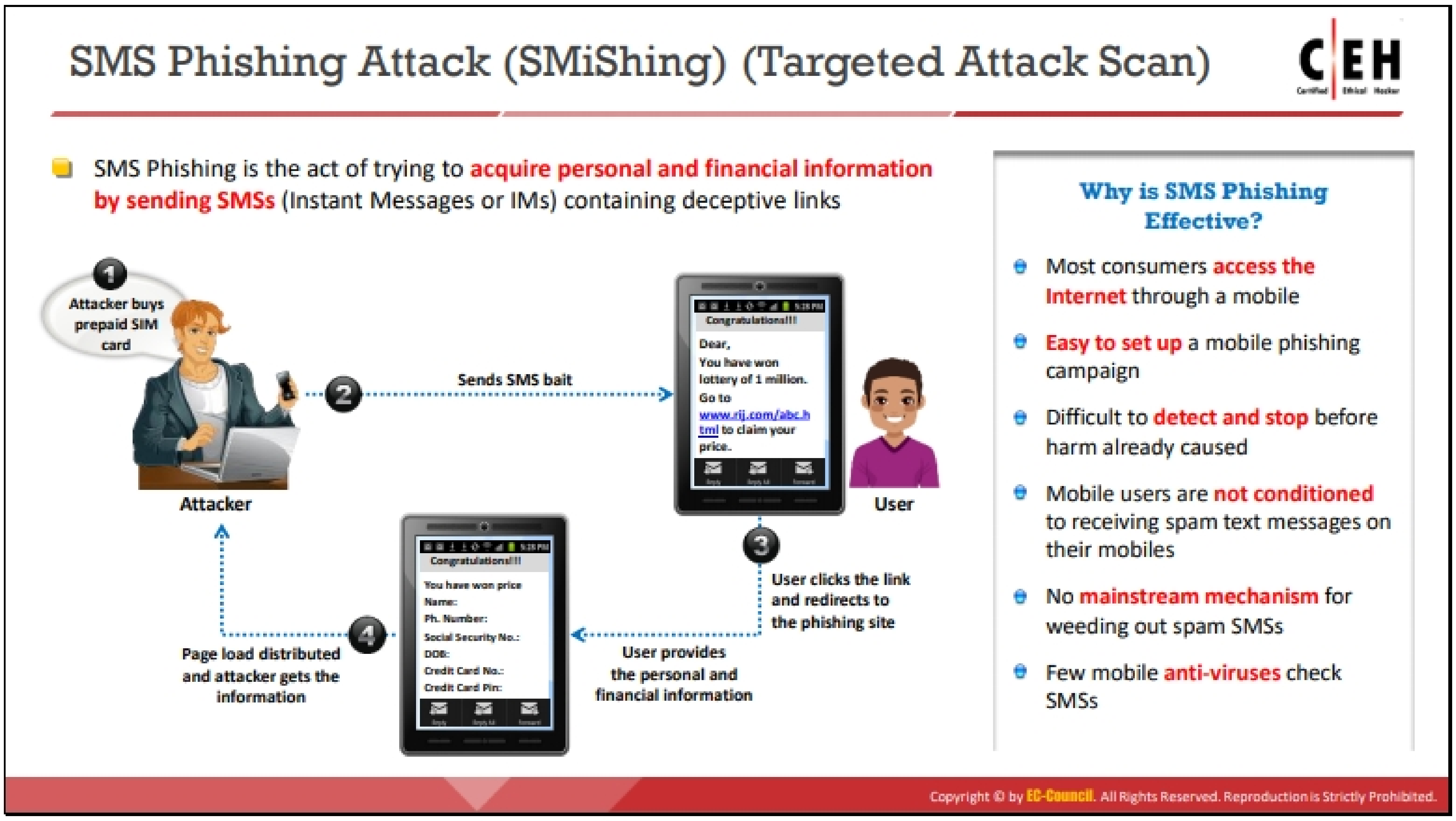
Congratulations you have won a $2000 Walmart Gift Card.

Click here to claim your gift.

www.WmarkClick.com/price.html

This is a generic email box. Please do not reply to this.
To unsubscribe, click this link to call our phone banking and register for do not disturb.``

Figure 17.5: Example of a spam message



SMS Phishing Attack (SMiShing) (Targeted Attack Scan)

- SMS Phishing is the act of trying to **acquire personal and financial information by sending SMSs** (Instant Messages or IMs) containing deceptive links

1 Attacker buys prepaid SIM card

Attacker

2 Sends SMS bait

Congratulations!!!

Dear,
You have won lottery of 1 million.
Go to www.rij.com/abc.html to claim your price.

User

3 User clicks the link and redirects to the phishing site

User provides the personal and financial information

Congratulations!!!

You have won price
Name:
Ph. Number:
Social Security No.:
DOB:
Credit Card No.:
Credit Card Pin:

4 Page load distributed and attacker gets the information

Why is SMS Phishing Effective?

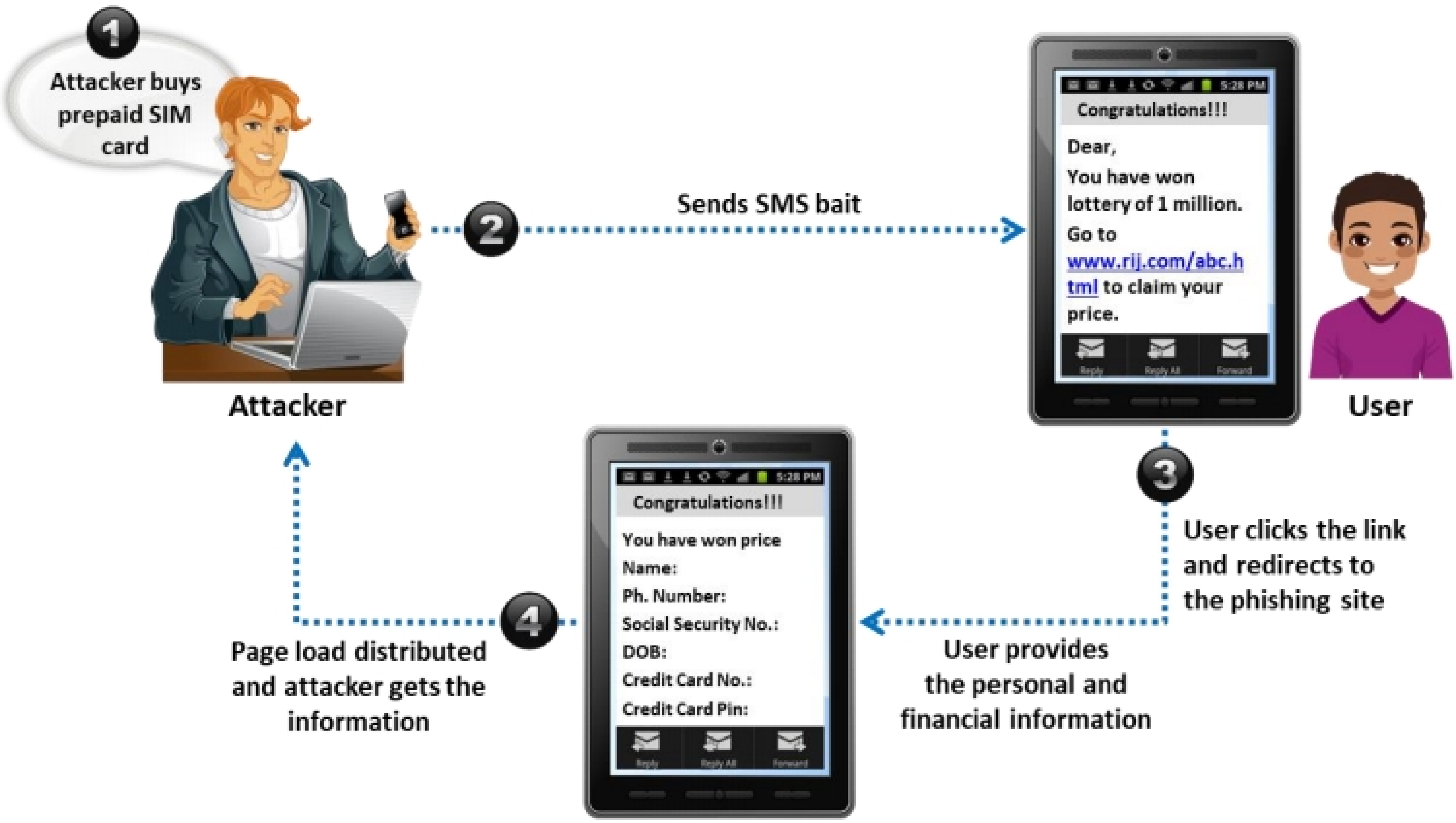
- Most consumers **access the Internet** through a mobile
- **Easy to set up** a mobile phishing campaign
- Difficult to **detect and stop** before harm already caused
- Mobile users are **not conditioned** to receiving spam text messages on their mobiles
- No **mainstream mechanism** for weeding out spam SMSs
- Few mobile **anti-viruses** check SMSs

SMS Phishing Attack (SMiShing) (Targeted Attack Scan)

Text messaging is the most prevalent nonvoice communication on mobile phones. Users around the world send and receive billions of text messages daily. Such a massive amount of data entails an increase in spam or phishing attacks.

SMS phishing (also known as SMiShing) is a type of phishing fraud in which an attacker uses SMS systems to send bogus text messages. It is the act of trying to acquire personal and financial information by sending SMS (or IM) containing deceptive links. Often, these bogus text messages contain a deceptive website URL or telephone number to lure victims into revealing their personal or financial information, such as SSNs, credit card numbers, and online banking username and password. In addition, attackers implement SMiShing to infect victims' mobile phones and associated networks with malware.

Attackers buy a prepaid SMS card using a fake identity. Then, they send an SMS bait to a user. The SMS may seem attractive or urgent. For example, it may include a lottery message, gift voucher, online purchase, or notification of account suspension, along with a malicious link or phone number. When the user clicks the link, considering it to be legitimate, he/she is redirected to the attacker's phishing site, where he/she provides the requested information (e.g., name, phone number, date of birth, credit card number or PIN, CVV code, SNN, and email address). The attacker may use the acquired information to perform malicious activities such as identity theft, online purchases, and so on.


1
Attacker buys prepaid SIM card
Attacker
2
Sends SMS bait
User
3
User clicks the link and redirects to the phishing site
User provides the personal and financial information
4
Page load distributed and attacker gets the information
5:28 PM
Reply
Reply All
Forward

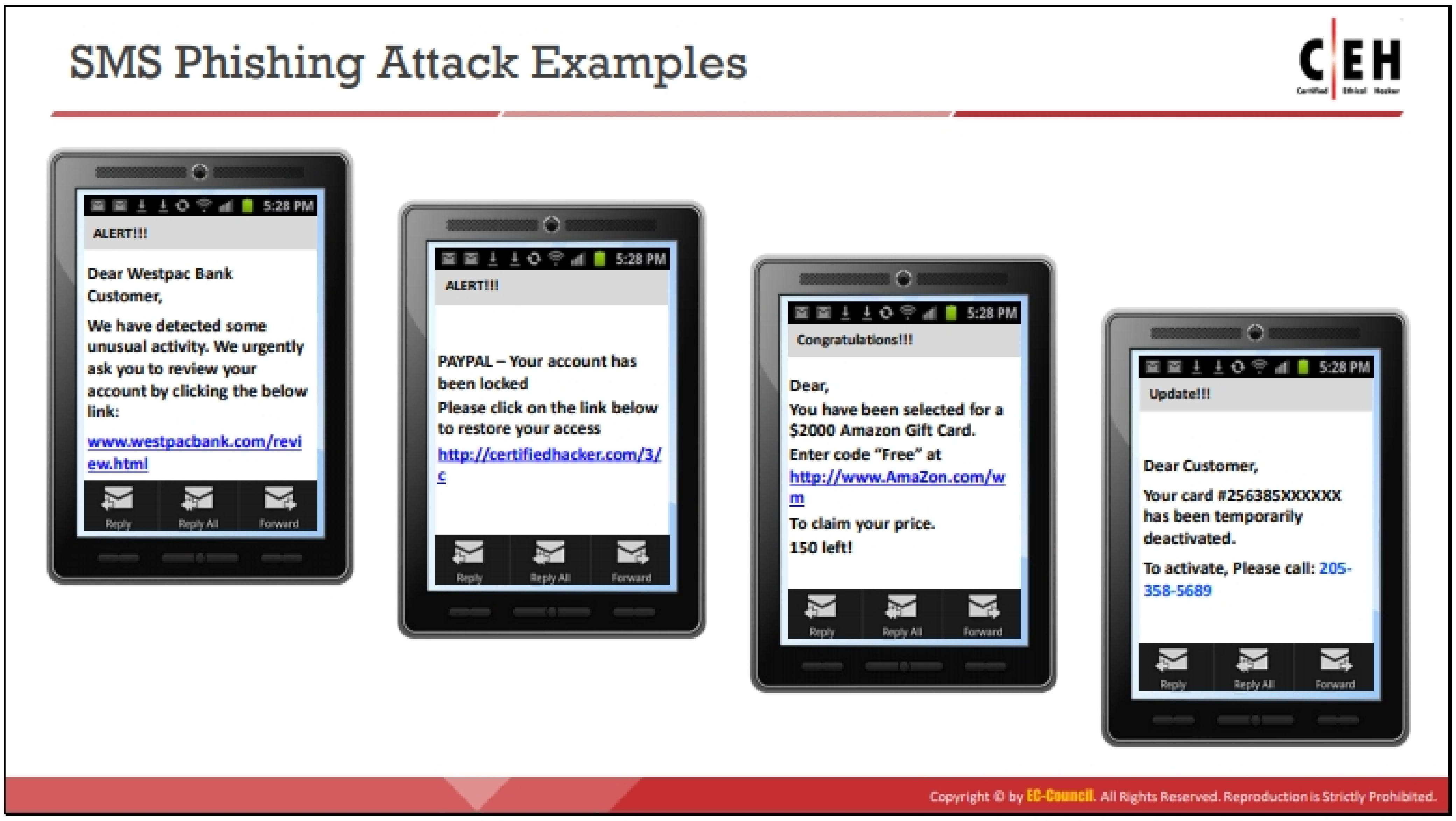
Congratulations!!!
Dear,
You have won lottery of 1 million.
Go to www.rij.com/abc.html to claim your price.

Congratulations!!!
You have won price
Name:
Ph. Number:
Social Security No.:
DOB:
Credit Card No.:
Credit Card Pin:

Figure 17.6: SMS Phishing process

Why is SMS Phishing Effective?

- Most consumers access the Internet through a mobile device.
- Easy to set up a mobile phishing campaign.
- Difficult to detect and stop it causes harm.
- Mobile users are not conditioned to receiving spam text messages on their mobile devices.
- No mainstream mechanism for weeding out spam SMS.
- Most mobile anti-virus tools do not check SMS.



SMS Phishing Attack Examples

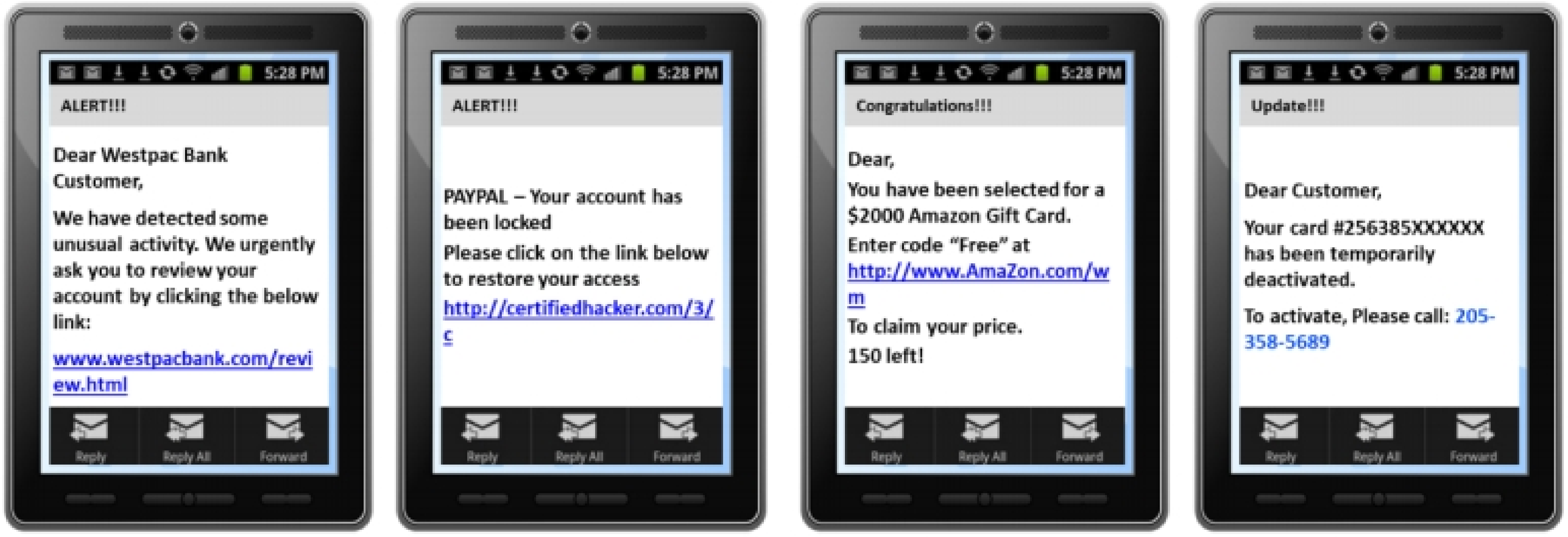
5:28 PM
ALERT!!!
Dear Westpac Bank Customer,
We have detected some unusual activity. We urgently ask you to review your account by clicking the below link:
www.westpacbank.com/review.html

5:28 PM
ALERT!!!
PAYPAL – Your account has been locked
Please click on the link below to restore your access
http://certifiedhacker.com/3/c

5:28 PM
Congratulations!!!
Dear,
You have been selected for a $2000 Amazon Gift Card.
Enter code "Free" at http://www.AmaZon.com/wm
To claim your price.
150 left!

5:28 PM
Update!!!
Dear Customer,
Your card #256385XXXXXX has been temporarily deactivated.
To activate, Please call: 205-358-5689

SMS Phishing Attack Examples



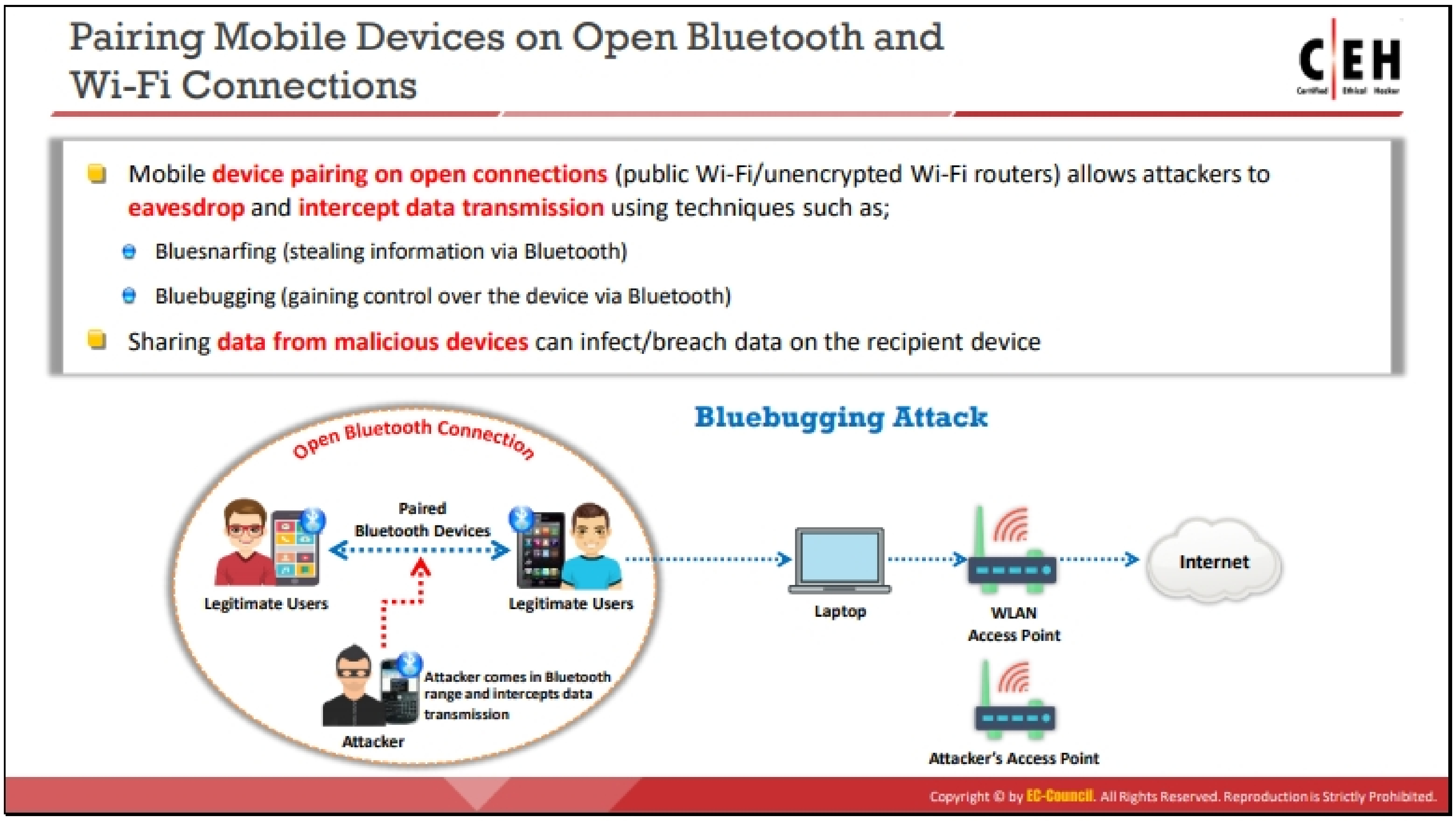
5:28 PM
ALERT!!!
Dear Westpac Bank Customer,
We have detected some unusual activity. We urgently ask you to review your account by clicking the below link:
www.westpacbank.com/review.html

5:28 PM
ALERT!!!
PAYPAL – Your account has been locked
Please click on the link below to restore your access
http://certifiedhacker.com/3/c

5:28 PM
Congratulations!!!
Dear,
You have been selected for a $2000 Amazon Gift Card.
Enter code "Free" at http://www.AmaZon.com/wm
To claim your price.
150 left!

5:28 PM
Update!!!
Dear Customer,
Your card #256385XXXXXX has been temporarily deactivated.
To activate, Please call: 205-358-5689

Figure 17.7: Examples of SMS Phishing



Pairing Mobile Devices on Open Bluetooth and Wi-Fi Connections

- Mobile **device pairing on open connections** (public Wi-Fi/unencrypted Wi-Fi routers) allows attackers to **eavesdrop** and **intercept data transmission** using techniques such as;
  - Bluesnarfing (stealing information via Bluetooth)
  - Bluebugging (gaining control over the device via Bluetooth)
- Sharing **data from malicious devices** can infect/breach data on the recipient device

Bluebugging Attack

Pairing Mobile Devices on Open Bluetooth and Wi-Fi Connections

Setting a mobile device's Bluetooth connection to "**open**" or the "**discovery**" mode and turning on the automatic Wi-Fi connection capability, particularly in public places, pose significant risks to mobile devices. Attackers exploit such settings to infect a mobile device with malware such as viruses and Trojans or compromise unencrypted data transmitted across untrusted networks. They may lure victims into accepting a Bluetooth connection request from a malicious device or they may perform a MITM attack to intercept and compromise all the data sent to and from the connected devices. Using the information gathered, attackers may engage in identity fraud and other malicious activities, thereby putting users at great risk.

Techniques such as "bluesnarfing" and "bluebugging" help an attacker to eavesdrop on or intercept data transmission between mobile devices paired on open connections (e.g., public Wi-Fi or unencrypted Wi-Fi routers).

- **Bluesnarfing** (Stealing information via Bluetooth)

  Bluesnarfing is the theft of information from a wireless device through a Bluetooth connection, often between phones, desktops, laptops, PDAs, and other devices. This technique allows an attacker to access the victim's contact list, emails, text messages, photos, videos, and business data, stored on the device.

  Any device with its Bluetooth connection enabled and set to "discoverable" (allowing other Bluetooth devices within range to view the device) may be susceptible to bluesnarfing if the vendor's software contains a certain vulnerability. Bluesnarfing exploits others' Bluetooth connections without their knowledge.

- **Bluebugging** (Taking over a device via Bluetooth)

  Bluebugging involves gaining remote access to a target Bluetooth-enabled device and using its features without the victim's knowledge or consent. Attackers compromise the target device's security to perform a backdoor attack prior to returning control to its owner. Bluebugging allows attackers to sniff sensitive corporate or personal data, receive calls and text messages intended for the victim, intercept phone calls and messages, forward calls and messages, connect to the Internet, and perform other malicious activities such as accessing contact lists, photos, and videos.

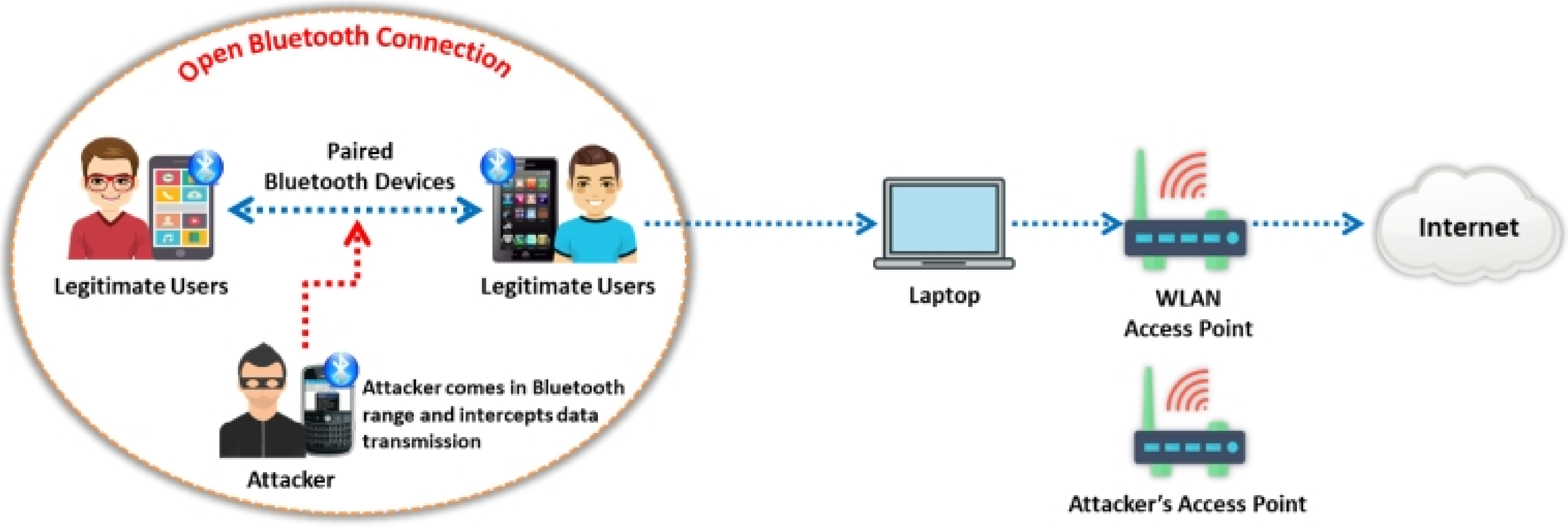
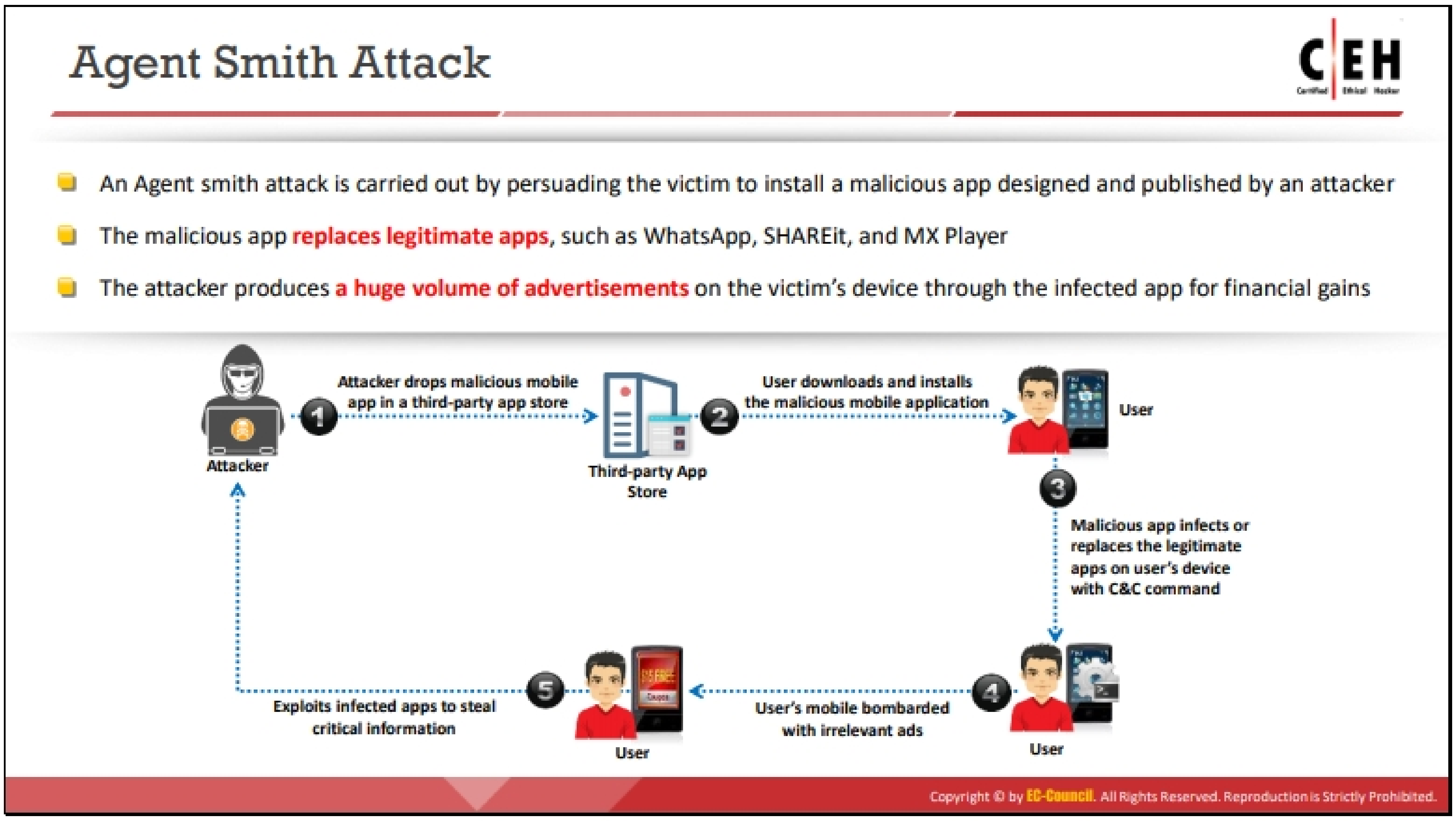

Open Bluetooth Connection
Paired
Bluetooth Devices
Legitimate Users
Legitimate Users
Attacker comes in Bluetooth range and intercepts data transmission
Attacker
Laptop
WLAN
Access Point
Internet
Attacker's Access Point

Figure 17.8: Bluebugging Attack



Agent Smith Attack

- An Agent smith attack is carried out by persuading the victim to install a malicious app designed and published by an attacker
- The malicious app **replaces legitimate apps**, such as WhatsApp, SHAREit, and MX Player
- The attacker produces **a huge volume of advertisements** on the victim's device through the infected app for financial gains

Attacker
1
Attacker drops malicious mobile app in a third-party app store
Third-party App Store
2
User downloads and installs the malicious mobile application
User
3
Malicious app infects or replaces the legitimate apps on user's device with C&C command
4
User
User's mobile bombarded with irrelevant ads
5
User
Exploits infected apps to steal critical information

Agent Smith Attack

Agent Smith attacks are carried out by luring victims into downloading and installing malicious apps designed and published by attackers in the form of games, photo editors, or other attractive tools from third-party app stores such as 9Apps. Once the user has installed the app, the core malicious code inside the application infects or replaces the legitimate apps in the victim's mobile device C&C commands. The deceptive application replaces legitimate apps such as WhatsApp, SHAREit, and MX Player with similar infected versions. The application sometimes also appears to be an authentic Google product such as Google Updater or Themes. The attacker then produces a massive volume of irrelevant and fraudulent advertisements on the victim's device through the infected app for financial gain. Attackers exploit these apps to steal critical information such as personal information, credentials, and bank details, from the victim's mobile device through C&C commands.

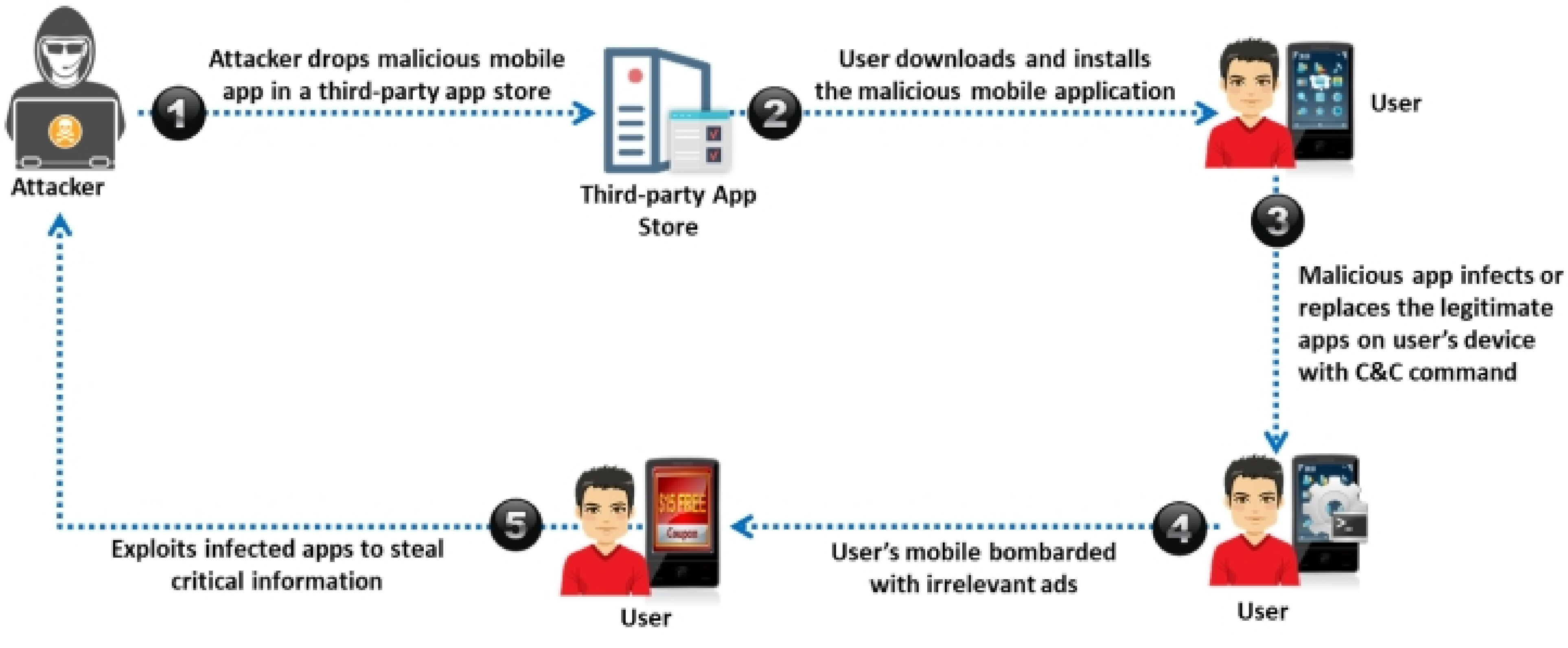
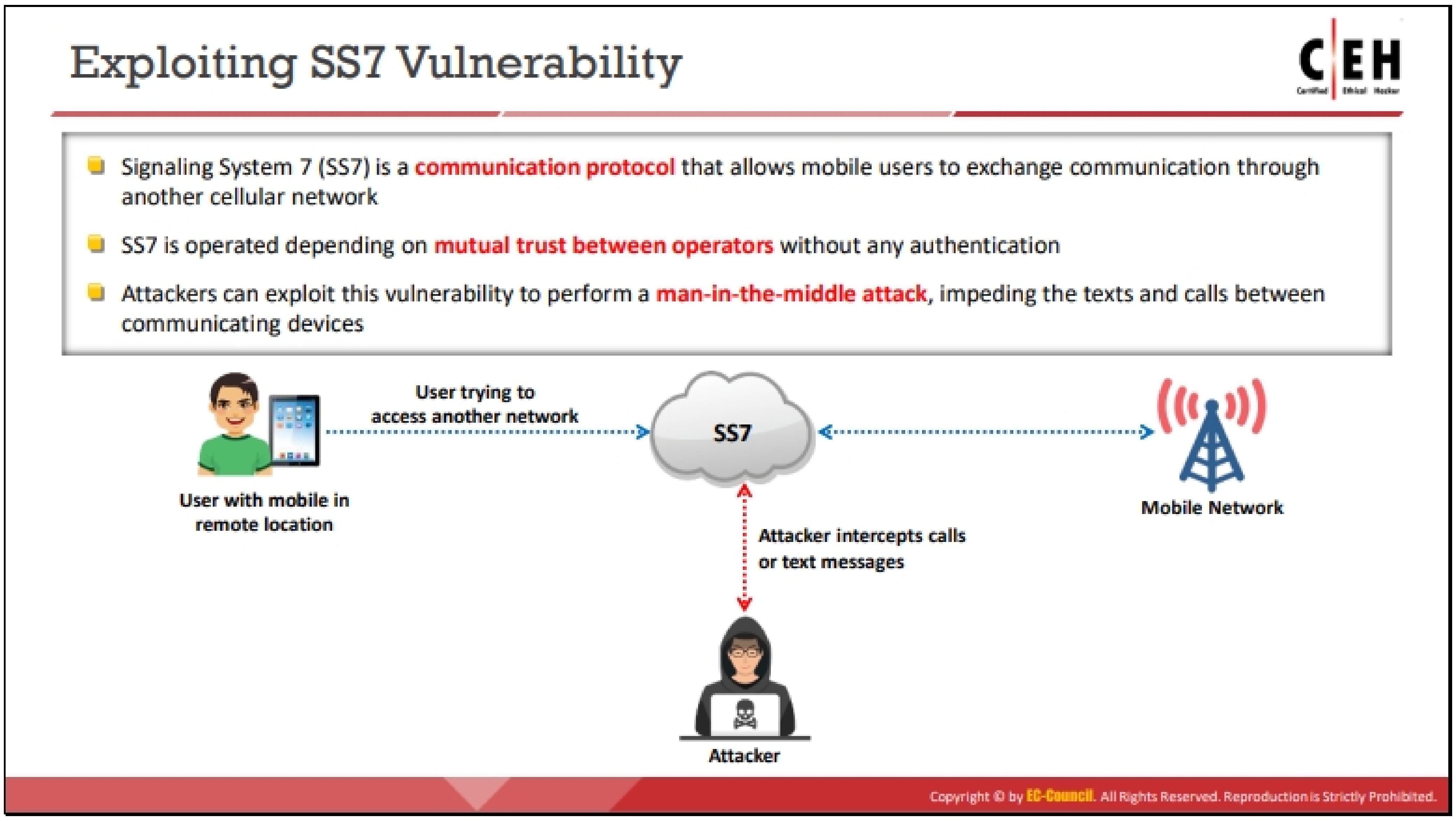

Attacker
1
Attacker drops malicious mobile app in a third-party app store
Third-party App Store
2
User downloads and installs the malicious mobile application
User
3
Malicious app infects or replaces the legitimate apps on user's device with C&C command
4
User
User's mobile bombarded with irrelevant ads
5
User
Exploits infected apps to steal critical information

Figure 17.9: Agent Smith Attack



Exploiting SS7 Vulnerability

- Signaling System 7 (SS7) is a **communication protocol** that allows mobile users to exchange communication through another cellular network
- SS7 is operated depending on **mutual trust between operators** without any authentication
- Attackers can exploit this vulnerability to perform a **man-in-the-middle attack**, impeding the texts and calls between communicating devices

User trying to access another network
SS7
User with mobile in remote location
Mobile Network
Attacker intercepts calls or text messages
Attacker

Exploiting SS7 Vulnerability

Signaling System 7 (SS7) is a communication protocol that allows mobile users to exchange communication through another cellular network (especially when roaming). Mobile devices are meant to be carried across different locations to serve their users. Changing the telecom operator or using the network of another cell tower is allowed via the SS7 protocol. This signaling mechanism is operated depending on mutual trust between the operators, without any authentication verification. Since the SS7 signaling network is not isolated, the attacker can exploit this vulnerability to perform an MITM attack by impeding text messages and calls between the communicating devices. The attacker can eavesdrop on bank credentials, OTPs and other sensitive information routed through the network. This vulnerability in SS7 can also allow the attacker to bypass two-factor authentication and end-to-end encryption via SMS.

Threats Associated with SS7 vulnerability

When the attacker gains access to the SS7 protocol, the victim's device faces the following risks:

- Exposing the subscriber's identity
- Revealing the network identity
- Spying on and intercepting the network to steal personal data
- Allowing phone tapping
- Performing DoS attacks to damage the reputation of the target telecom operator
- Tracking geographic locations

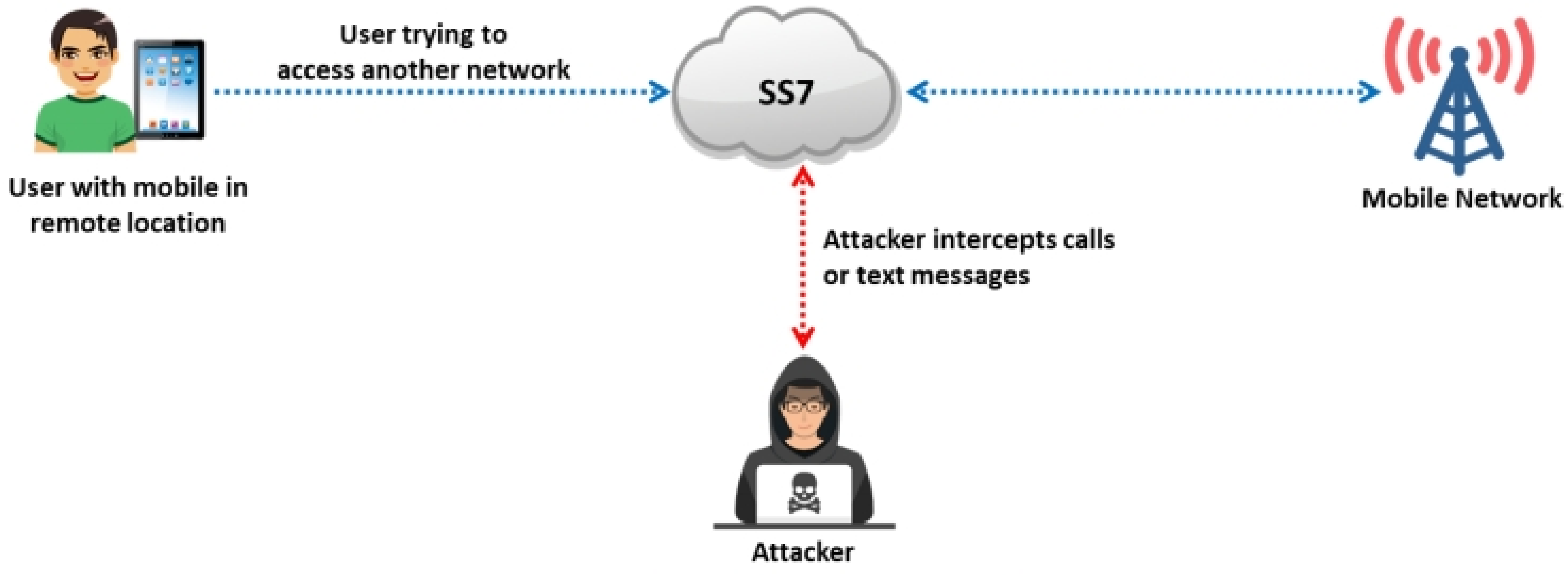
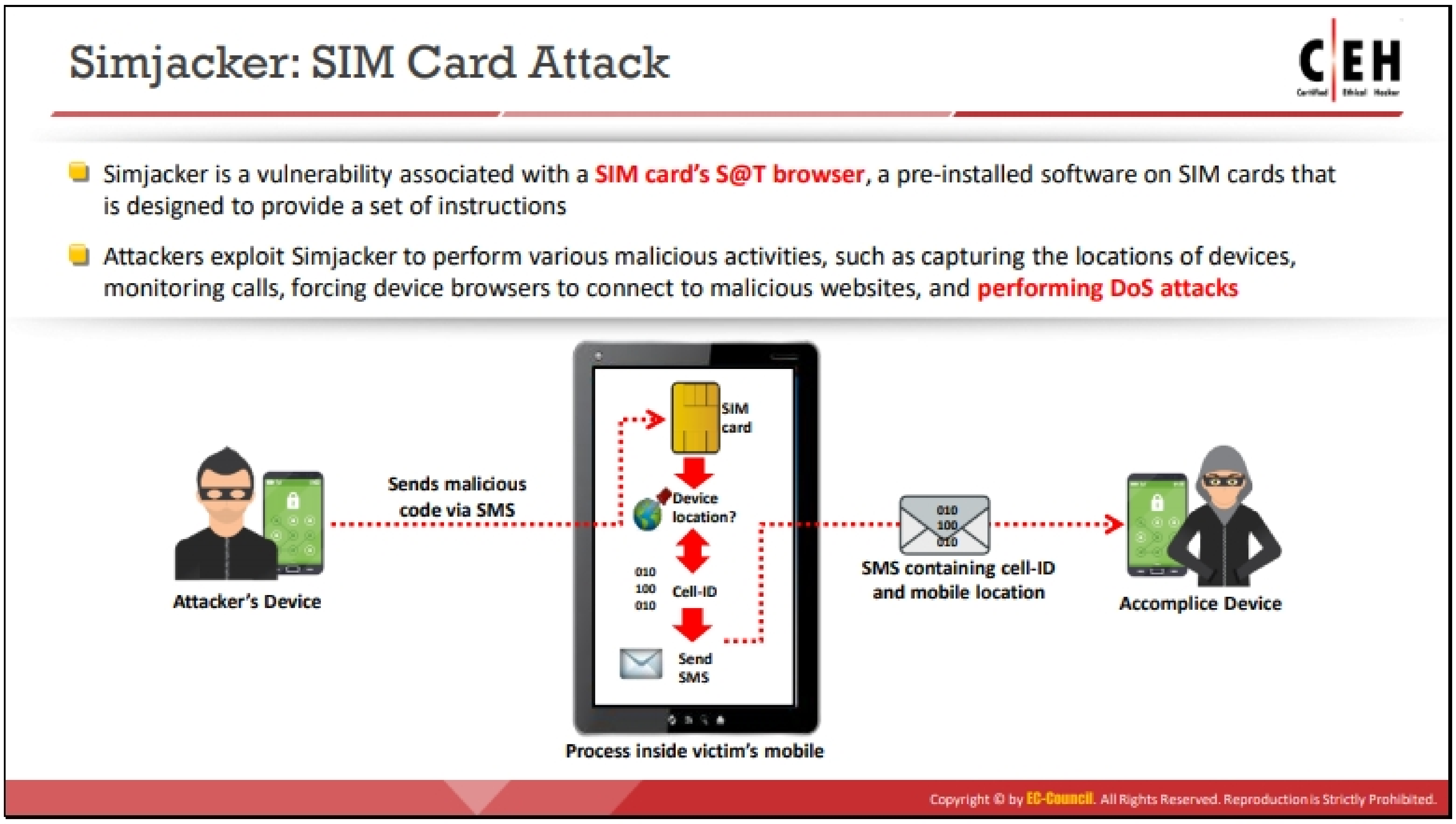

User trying to
access another network
SS7
User with mobile in
remote location
Mobile Network
Attacker intercepts calls
or text messages
Attacker

Figure 17.10: Exploiting SS7 vulnerability



Simjacker: SIM Card Attack

- Simjacker is a vulnerability associated with a **SIM card's S@T browser**, a pre-installed software on SIM cards that is designed to provide a set of instructions
- Attackers exploit Simjacker to perform various malicious activities, such as capturing the locations of devices, monitoring calls, forcing device browsers to connect to malicious websites, and **performing DoS attacks**

Attacker's Device
Sends malicious code via SMS
SIM card
Device location?
010 100 010
Cell-ID
Send SMS
Process inside victim's mobile
010 100 010
SMS containing cell-ID and mobile location
Accomplice Device

Simjacker: SIM Card Attack

Simjacker is a vulnerability associated with a SIM card's S@T browser (SIMalliance Toolbox Browser), a pre-installed software incorporated in SIM cards to provide a set of instructions. Attackers exploit this vulnerability in the S@T browser to perform various malicious activities such as capturing the device location, monitoring calls, gathering information such as IMEI, making fraudulent or expensive calls, sending premium-rate messages, forcing the device browser to connect to malicious websites, and performing DoS attacks to block SIM cards. The SIM card-based attack can be aggravated based on the victim's device. The Simjacker attack is initiated by sending spyware-like code in the form of system or SIM card settings through an SMS to take complete control of the SIM card and mobile device to issue various commands without user interaction.

Steps involved in Simjacker attack

- The attacker sends fraudulent SMS containing hidden code or instructions from a SIM Application Toolkit (STK)
- The victim receives the malicious SMS and the S@T browser on the SIM card automatically recognizes and processes the hidden instructions or code
- The injected code performs various activities on the device without the user's consent
- The accomplice device receives the user information via SMS, which an attacker can use to track live locations, exfiltrate the device information, and perform many other malicious activities

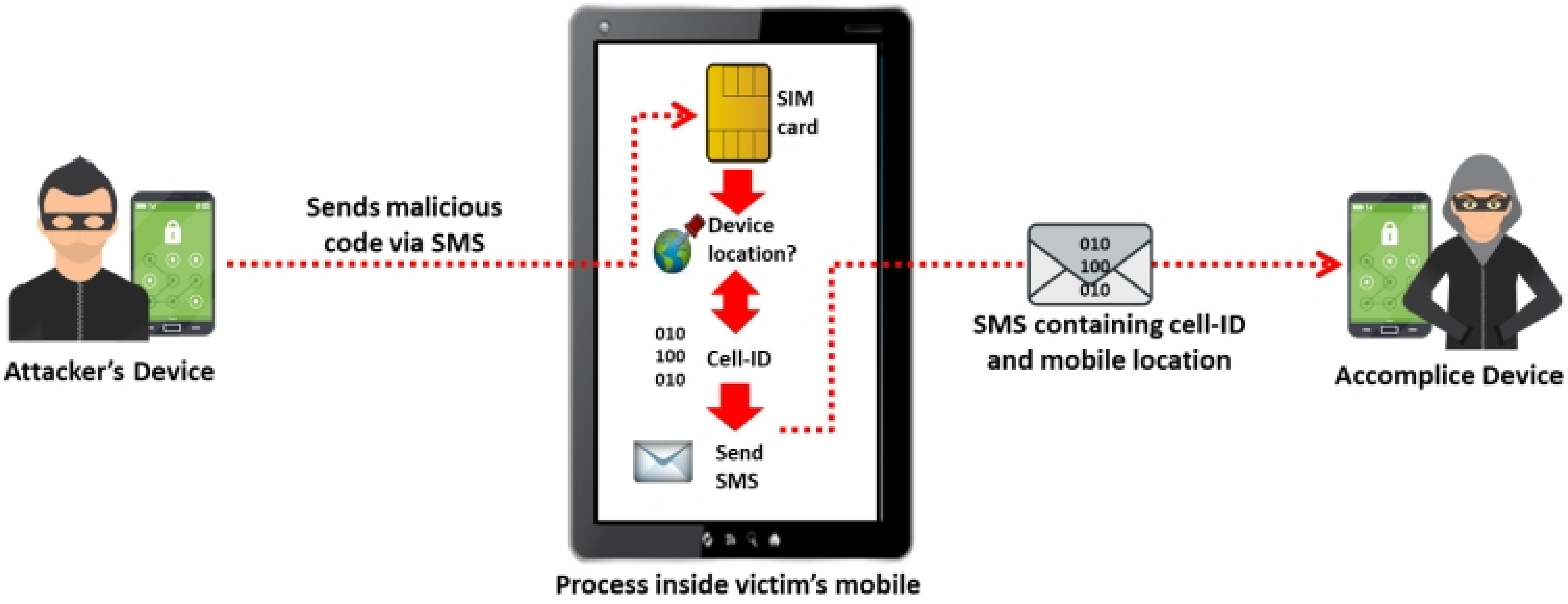
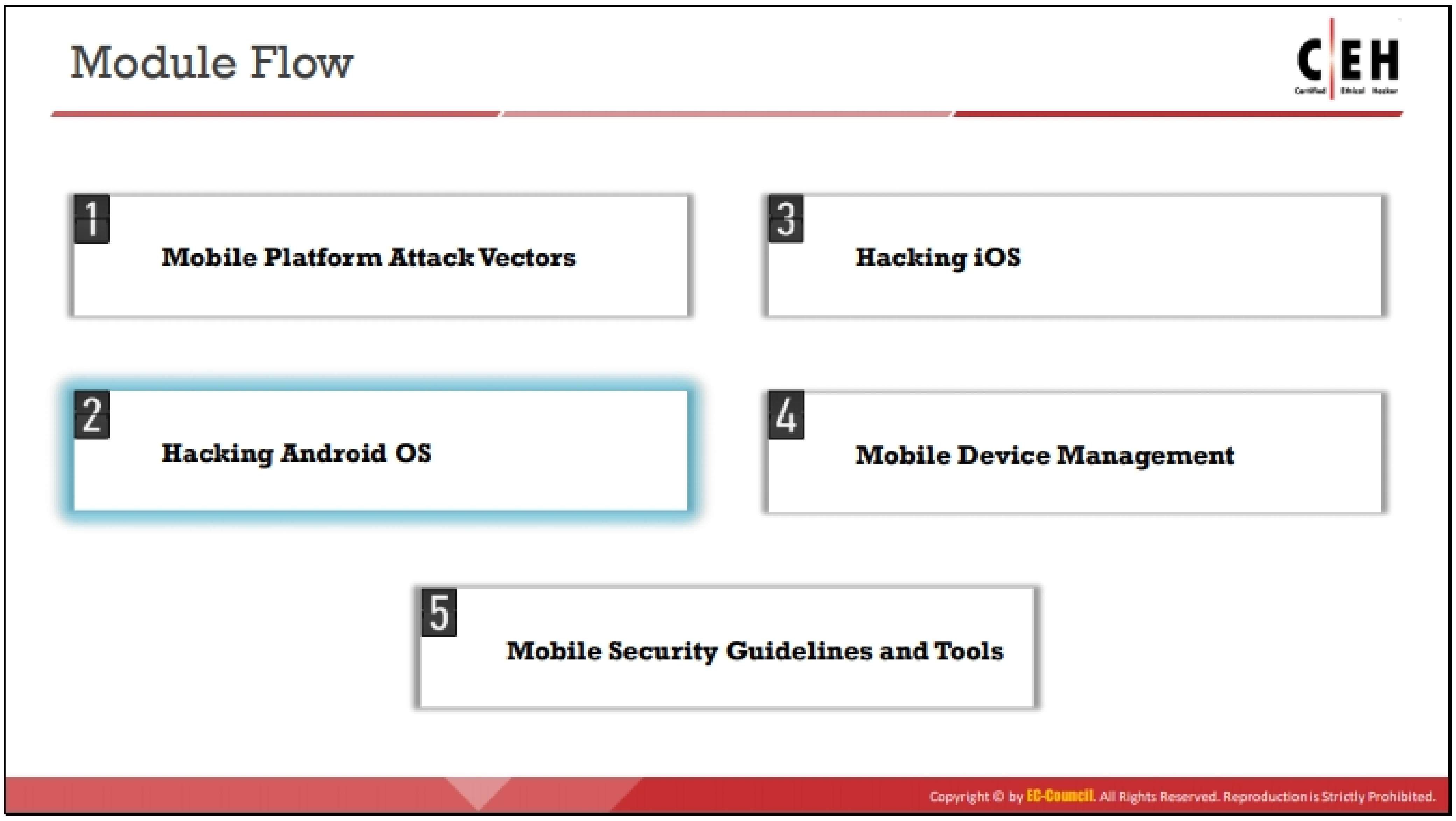

Attacker's Device
Sends malicious code via SMS
SIM card
Device location?
010 100 010
Cell-ID
Send SMS
Process inside victim's mobile
010 100 010
SMS containing cell-ID and mobile location
Accomplice Device

Figure 17.11: Exploiting Simjacker vulnerability



Module Flow

1 **Mobile Platform Attack Vectors**

2 **Hacking Android OS**

3 **Hacking iOS**

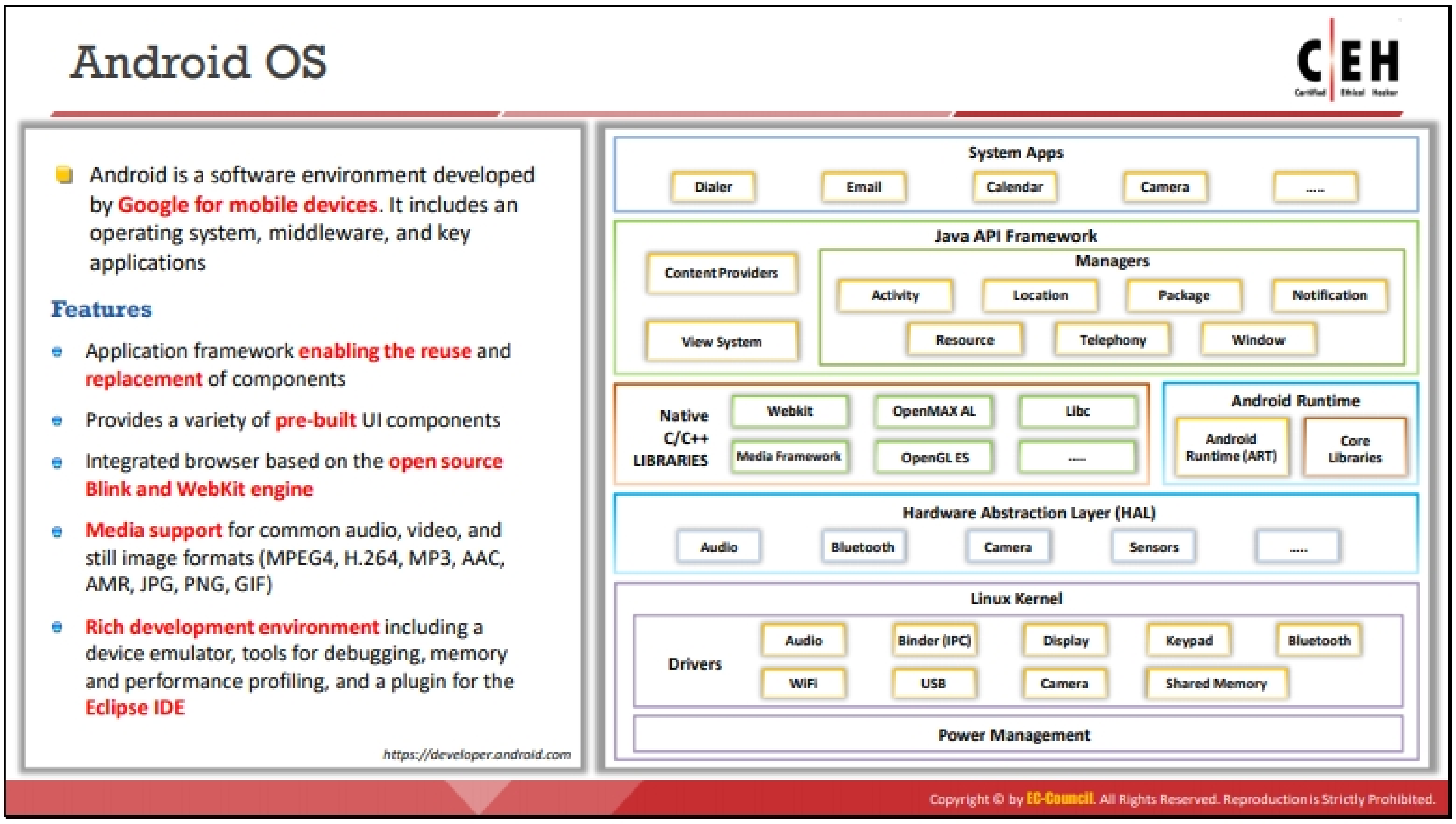
4 **Mobile Device Management**

5 **Mobile Security Guidelines and Tools**

Hacking Android OS

The number of people using smartphones and tablets is increasing rapidly, as these devices support a wide range of functionalities. Android is the most popular mobile OS because it is a platform that is open to all applications. Like other OSs, Android has certain vulnerabilities, and not all Android users install patches to update and secure the OS software and apps. Such a casual approach of users allows attackers to exploit vulnerabilities and launch various types of attacks to steal valuable data stored on the victims' devices.

This section discusses the Android OS, its architecture, and the associated vulnerabilities. It also covers the process of rooting Android phones, rooting tools, Android Trojans, and hacking Android mobiles. Finally, the section discusses guidelines for securing Android devices, security controls, and device-tracking tools.



Android OS

- Android is a software environment developed by **Google for mobile devices**. It includes an operating system, middleware, and key applications

Features

- Application framework **enabling the reuse** and **replacement** of components
- Provides a variety of **pre-built** UI components
- Integrated browser based on the **open source Blink and WebKit engine**
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the **Eclipse IDE**

https://developer.android.com

System Apps
Dialer | Email | Calendar | Camera | .....

Java API Framework
Content Providers | View System
Managers: Activity | Location | Package | Notification | Resource | Telephony | Window

Native C/C++ LIBRARIES
Webkit | OpenMAX AL | Libc | Media Framework | OpenGL ES | .....

Android Runtime
Android Runtime (ART) | Core Libraries

Hardware Abstraction Layer (HAL)
Audio | Bluetooth | Camera | Sensors | .....

Linux Kernel
Drivers: Audio | Binder (IPC) | Display | Keypad | Bluetooth | WiFi | USB | Camera | Shared Memory
Power Management

Android OS

Source: *https://developer.android.com*

Android, a software environment developed by Google for mobile devices, includes an OS, middleware, and key applications. The Android OS relies on the Linux kernel and is an open-source platform.

Features:

- Provides a variety of prebuilt UI components such as structured layout objects and UI controls that allow one to build the GUI for the app
- Provides several options to save persistent application data:
  - **Shared Preferences**—Store private primitive data in key-value pairs
  - **Internal Storage**—Private data on the device memory
  - **External Storage**—Public data on the shared external storage
  - **SQLite Databases**—Store structured data in a private database
  - **Network Connection**—Store data on the web with your own network server
- RenderScript provides a platform-independent computation engine that operates at the native level. One can use it to accelerate apps that require extensive computational horsepower.
- Provides rich APIs that allow the app to connect and interact with other devices over Bluetooth, near-field communication (NFC), Wi-Fi P2P, USB, and session initiation protocol (SIP), in addition to standard network connections.

- Application framework allows for the reuse and replacement of components.
- Android runtime (ART) optimized for mobile devices.
- Integrated browser based on the open-source Blink and WebKit engine.
- SQLite for structured data storage.
- Media support for common audio, video, and still image formats (e.g., MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, and GIF).
- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE.

Android OS Architecture

Source: *https://developer.android.com*

Android is a Linux-based OS designed for portable devices such as smartphones and tablets. It is a stack of software components categorized into six sections (System Apps, Java AP Framework, Native C/C++ Libraries, Android Runtime, Hardware Abstraction Layer (HAL), and Linux kernel) and five layers.

- **System Apps**

  All Android system applications are at the top layer. Any app developed should fit into this layer. Some standard applications that come pre-installed with every Android device include dialer, email, calendar, camera, SMS messaging, web browsers, contact managers, and so on. Most Android apps are “written” in Java.

- **Java API Framework**

  Android platform functions are made available to developers through APIs written in Java. The application framework offers many high-level services to applications, which developers incorporate in their development.

  Some of the application framework blocks are as follows:

  - **Content Providers**—Manages data sharing between applications.
  - **View System**—For developing lists, grids, text boxes, buttons, and so on.
  - **Activity Manager**—Controls the activity life cycle of applications.
  - **Location Manager**—Manages location using GPS or cell towers.
  - **Package Manager**—Keeps track of the applications installed on the device.
  - **Notification Manager**—Helps applications display custom messages in a status bar.
  - **Resource Manager**—Manages various types of resources used.
  - **Telephony Manager**—Manages all voice calls.
  - **Window Manager**—Manages application windows.

- **Native C/C++ Libraries**

  The next layer comprises the native libraries. Libraries are “written” in C or C++ and are specific to particular hardware. This layer allows the device to control different types of data.

  The native libraries are as follows:

  - **WebKit and Blink**—web browser engine to display HTML content
  - **Open Max AL**—companion API to OpenGL ES but used for multimedia (video and audio) rather than audio only
  - **Libc**—Comprises System C libraries
  - **Media Framework**—provides media codecs that allow recording and playback of different media formats
  - **Open GL | ES**—2D and 3D graphics library
  - **Surface Manager**—meant for display management
  - **SQLite**—database engine used for data storage purposes
  - **FreeType**—meant for rendering fonts
  - **SSL**—meant for Internet security

- **Android Runtime**

  It includes core libraries and the ART virtual machine.

  - **Android Runtime (ART)**

    For Android versions beyond 5.0, apps have their own runtime processes and instances. Android runtime has features such as ahead-of-time (AOT) compilation, just-in-time (JIT) compilation, optimized garbage collection (GC), and Dalvik Executable format (DEX) files to compress machine code.

  - **Core Libraries**

    The set of core libraries allows developers to write Android applications using Java.

- **Hardware Abstraction Layer**

  The hardware abstraction layer is used to expose the device’s hardware capabilities to the Java API framework that resides at a higher level. It acts as an abstraction layer between the hardware and the software stack. HAL comprises various modules that are required for the hardware equipment in the device, such as audio, camera, Bluetooth, sensors, and so on.

- **Linux Kernel**

  The Android OS relies on the Linux kernel. This layer comprises low-level device drivers such as audio driver, binder (IPC) driver, display driver, keypad driver, Bluetooth driver, camera driver, shared memory driver, USB driver, Wi-Fi driver, Flash memory driver, and power management for the various hardware components. The functions of this layer include memory management, power management, security management, and networking.

The figure below shows a pictorial representation of the complete Android architecture:

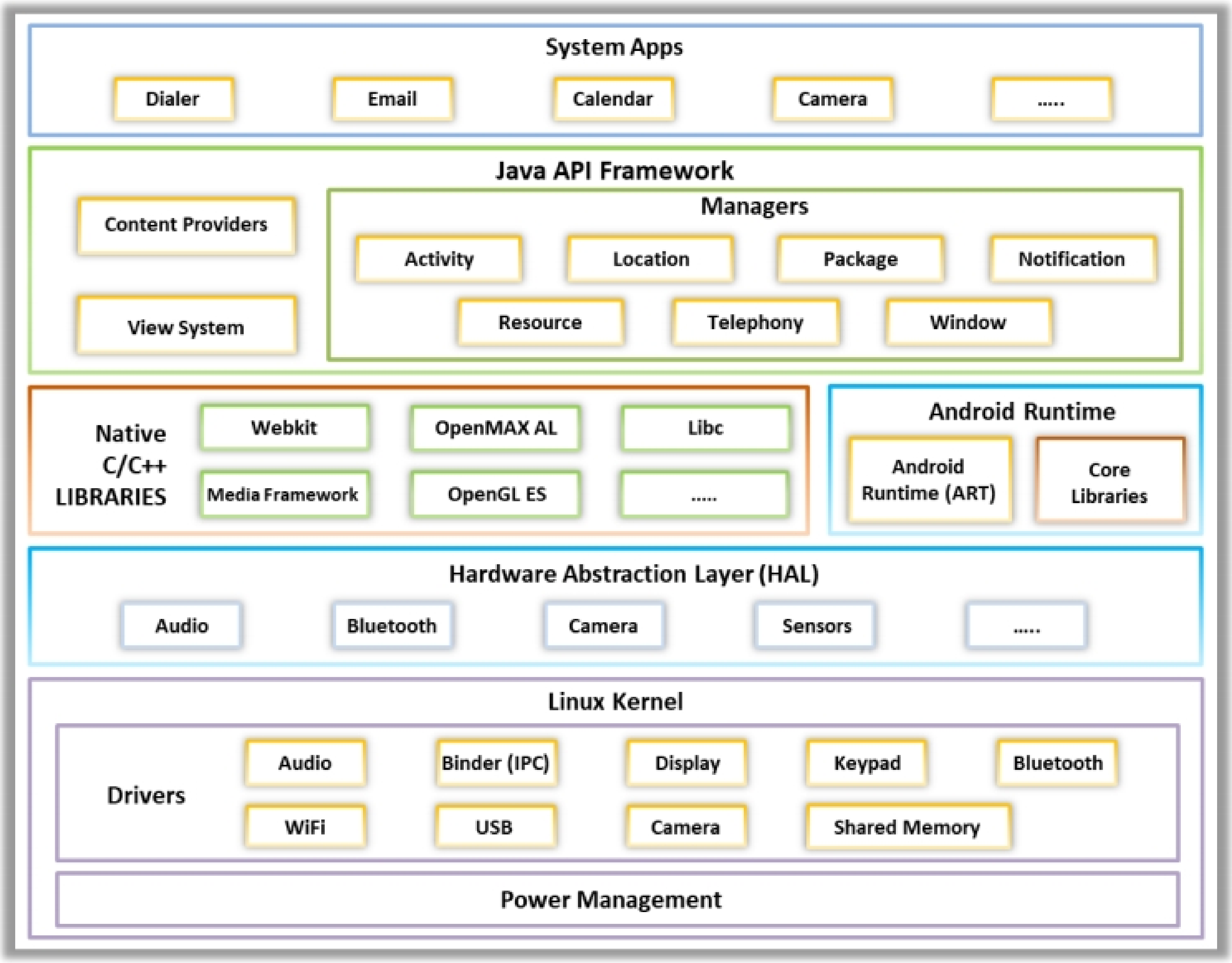
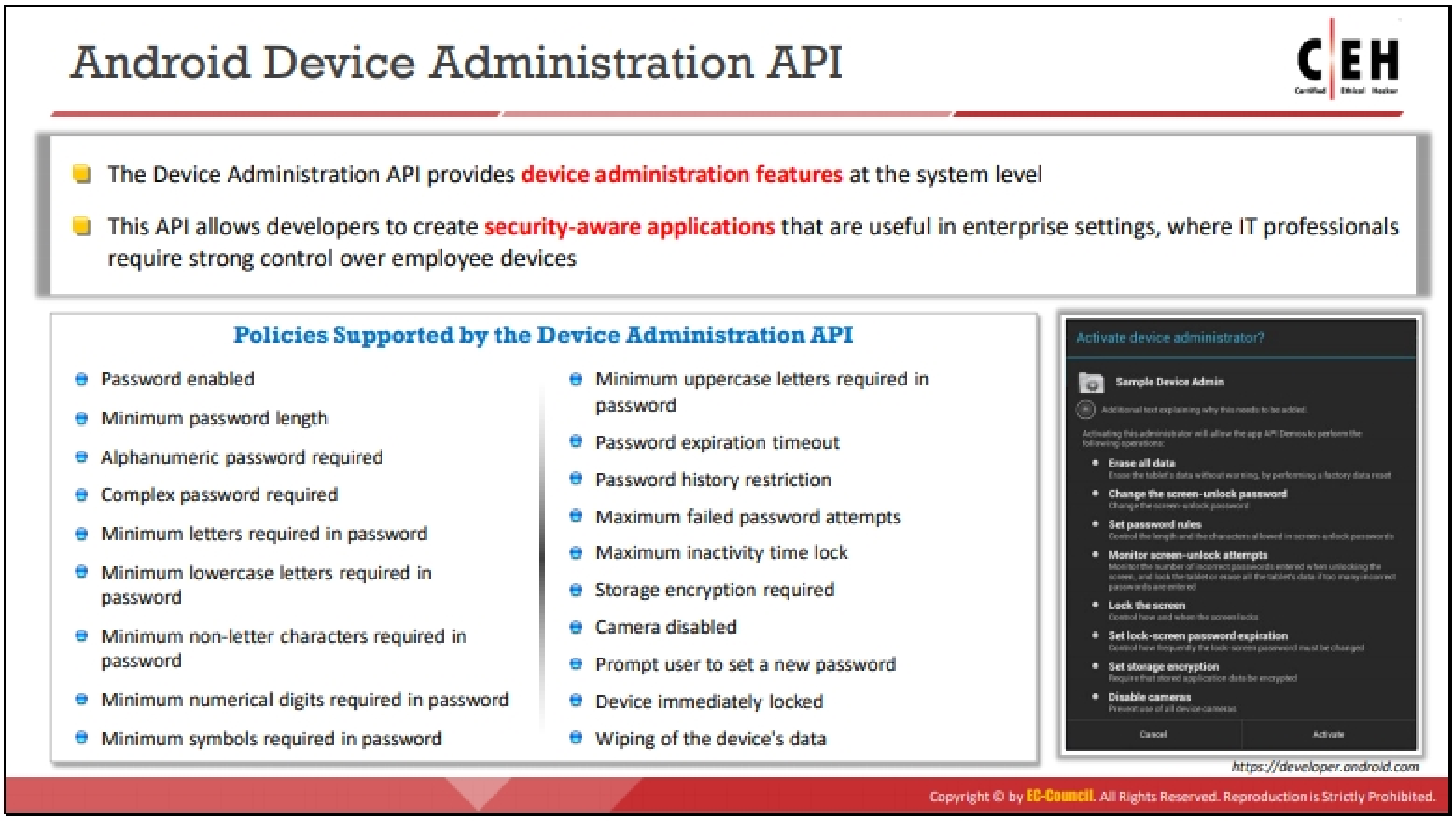

System Apps
Dialer
Email
Calendar
Camera
.....
Java API Framework
Content Providers
View System
Managers
Activity
Location
Package
Notification
Resource
Telephony
Window
Native C/C++ LIBRARIES
Webkit
OpenMAX AL
Libc
Media Framework
OpenGL ES
.....
Android Runtime
Android Runtime (ART)
Core Libraries
Hardware Abstraction Layer (HAL)
Audio
Bluetooth
Camera
Sensors
.....
Linux Kernel
Drivers
Audio
Binder (IPC)
Display
Keypad
Bluetooth
WiFi
USB
Camera
Shared Memory
Power Management

Figure 17.12: Android Architecture



Android Device Administration API

- The Device Administration API provides **device administration features** at the system level
- This API allows developers to create **security-aware applications** that are useful in enterprise settings, where IT professionals require strong control over employee devices

Policies Supported by the Device Administration API

- Password enabled
- Minimum password length
- Alphanumeric password required
- Complex password required
- Minimum letters required in password
- Minimum lowercase letters required in password
- Minimum non-letter characters required in password
- Minimum numerical digits required in password
- Minimum symbols required in password
- Minimum uppercase letters required in password
- Password expiration timeout
- Password history restriction
- Maximum failed password attempts
- Maximum inactivity time lock
- Storage encryption required
- Camera disabled
- Prompt user to set a new password
- Device immediately locked
- Wiping of the device's data

Activate device administrator?

Sample Device Admin

Additional text explaining why this needs to be added.

Activating this administrator will allow the app API Demos to perform the following operations:

- **Erase all data**
  Erase the tablet's data without warning, by performing a factory data reset
- **Change the screen-unlock password**
  Change the screen-unlock password
- **Set password rules**
  Control the length and the characters allowed in screen-unlock passwords
- **Monitor screen-unlock attempts**
  Monitor the number of incorrect passwords entered when unlocking the screen, and lock the tablet or erase all the tablet's data if too many incorrect passwords are entered
- **Lock the screen**
  Control how and when the screen locks
- **Set lock-screen password expiration**
  Control how frequently the lock-screen password must be changed
- **Set storage encryption**
  Require that stored application data be encrypted
- **Disable cameras**
  Prevent use of all device cameras

https://developer.android.com

Android Device Administration API

Source: *https://developer.android.com*

The device administration API provides device administration features at the system level. Such APIs allow developers to create security-aware applications that are useful in enterprise settings, in which IT professionals require rich control over employee devices. One can use a device administration ("admin") API to write device admin applications that users install on their devices. The device admin application enforces the desired policies.

Some examples of the types of applications that might use the device administration API are as follows:

- Email clients
- Security applications that perform a remote wipe
- Device management services and applications

The table below lists the policies supported by the Android device administration API:

Policy	Description
Password enabled	Requires that devices ask for a PIN or password
Minimum password length	Set the required number of characters for the password. For example, you can require a PIN or password to have at least six characters.
Alphanumeric password required	Requires the password to have a combination of letters and numbers and may include symbolic characters.
Complex password	Requires the password to contain at least a letter, a numerical digit,

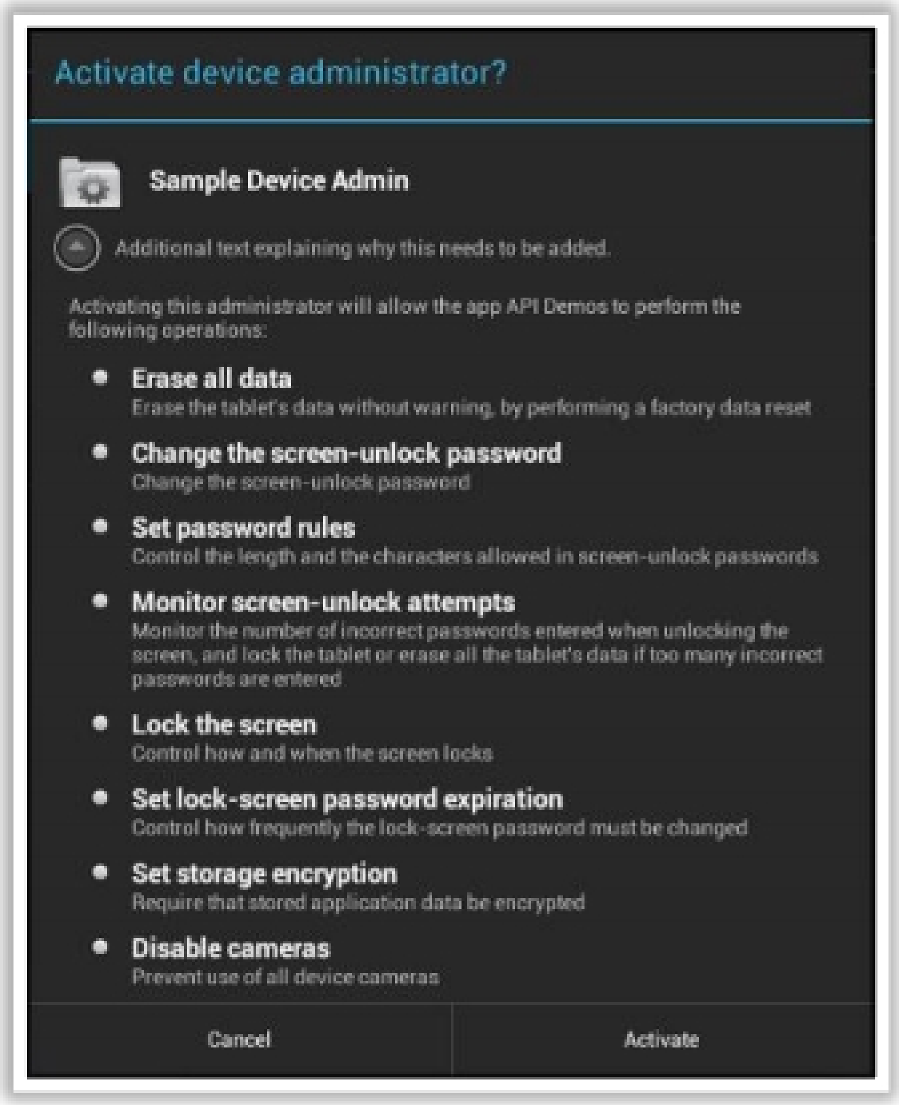
required	and a special symbol. Introduced in Android 3.0.
Minimum letters required in password	The minimum number of letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum lowercase letters required in password	The minimum number of lowercase letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum non-letter characters required in password	The minimum number of nonletter characters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum numerical digits required in password	The minimum number of numerical digits required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum symbols required in password	The minimum number of symbols required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum uppercase letters required in password	The minimum number of uppercase letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Password expiration timeout	When the password will expire, expressed as a delta in milliseconds from when a device admin sets the expiration timeout. Introduced in Android 3.0.
Password history restriction	This policy prevents users from reusing the last *n* unique passwords. Typically, you can use this policy in conjunction with setPasswordExpirationTimeout(), which forces users to update their passwords after a specified amount of time has elapsed. Introduced in Android 3.0.
Maximum failed password attempts	Specifies how many times a user can enter the wrong password before the device wipes its data. The Device Administration API also allows administrators to remotely reset the device to factory defaults. This secures data in case the device is lost or stolen.
Maximum inactivity time lock	Sets the length of time since the user last touched the screen or pressed a button before the device locks the screen. When this happens, users need to enter their PIN or password again before they can use their devices and access data. The value can be between 1 and 60 minutes.
Require storage encryption	Specifies the encryption of storage, if the device supports it. Introduced in Android 3.0.
Disable camera	Specifies the camera-disabling feature. Note that this does not have to be permanent. The camera can be enabled/ disabled dynamically based on context, time, and so on. Introduced in Android 4.0.

Table 17.3: List of policies supported by the Android Device Administration API

In addition to supporting the policies mentioned above, the device administration API lets you perform the following:

- Prompt user to set a new password
- Lock device immediately
- Wipe the device's data (i.e., restore the device to its factory defaults)

An example of an Android device administrator page is shown below:



Activate device administrator?

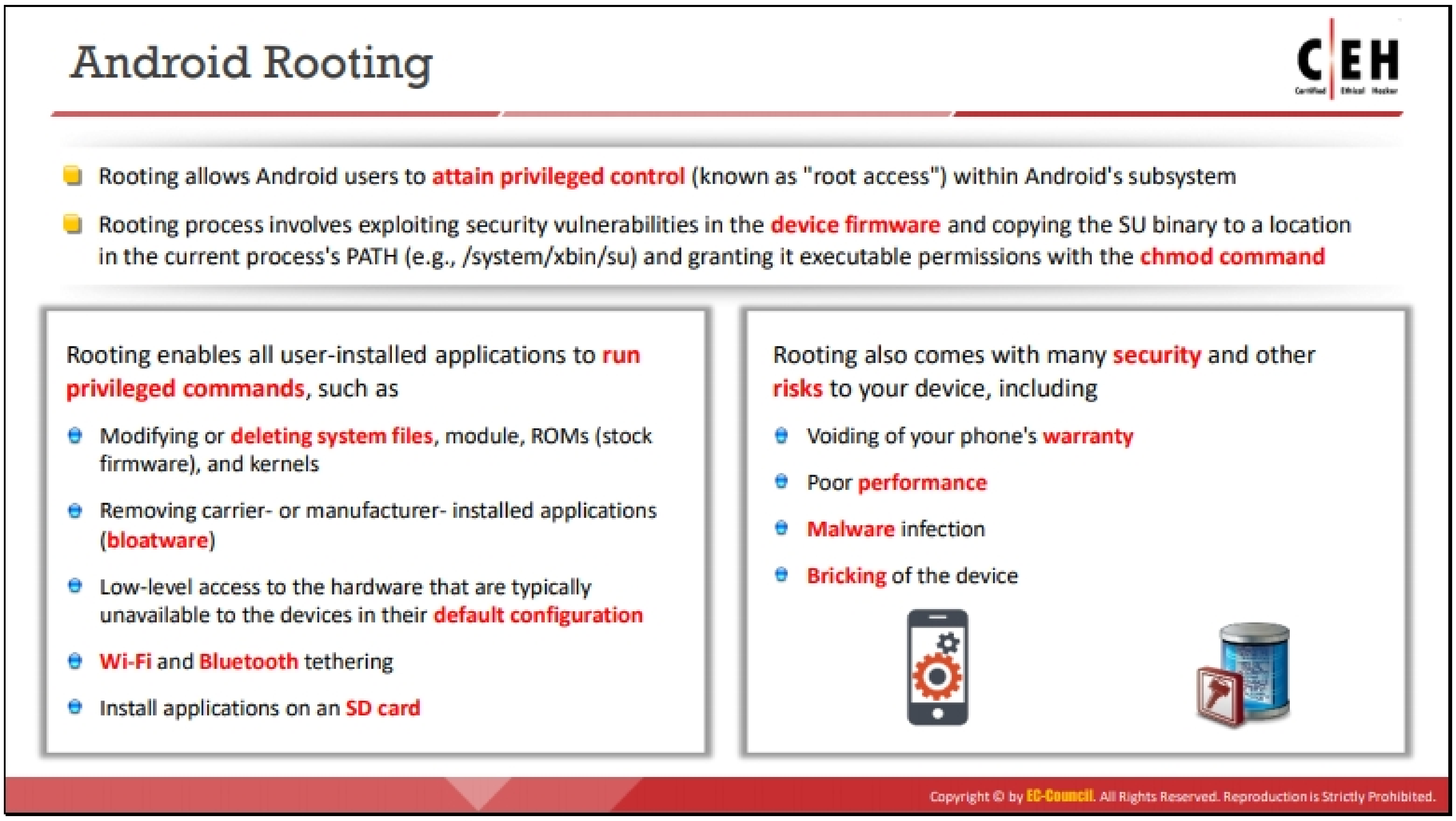
Sample Device Admin

Additional text explaining why this needs to be added.

Activating this administrator will allow the app API Demos to perform the following operations:

- **Erase all data**
  Erase the tablet's data without warning, by performing a factory data reset
- **Change the screen-unlock password**
  Change the screen-unlock password
- **Set password rules**
  Control the length and the characters allowed in screen-unlock passwords
- **Monitor screen-unlock attempts**
  Monitor the number of incorrect passwords entered when unlocking the screen, and lock the tablet or erase all the tablet's data if too many incorrect passwords are entered
- **Lock the screen**
  Control how and when the screen locks
- **Set lock-screen password expiration**
  Control how frequently the lock-screen password must be changed
- **Set storage encryption**
  Require that stored application data be encrypted
- **Disable cameras**
  Prevent use of all device cameras

Figure 17.13: Screenshot displaying Android Device Administrator



Android Rooting

- Rooting allows Android users to **attain privileged control** (known as "root access") within Android's subsystem
- Rooting process involves exploiting security vulnerabilities in the **device firmware** and copying the SU binary to a location in the current process's PATH (e.g., /system/xbin/su) and granting it executable permissions with the **chmod command**

Rooting enables all user-installed applications to **run privileged commands**, such as

- Modifying or **deleting system files**, module, ROMs (stock firmware), and kernels
- Removing carrier- or manufacturer- installed applications (**bloatware**)
- Low-level access to the hardware that are typically unavailable to the devices in their **default configuration**
- **Wi-Fi** and **Bluetooth** tethering
- Install applications on an **SD card**

Rooting also comes with many **security** and other **risks** to your device, including

- Voiding of your phone's **warranty**
- Poor **performance**
- **Malware** infection
- **Bricking** of the device

Android Rooting

The goal of rooting Android is to overcome the restrictions imposed by hardware manufacturers and carriers, thereby resulting in the ability to modify or replace system applications and settings, run apps that require admin privileges, remove and replace a device's OS, remove applications pre-installed by its manufacturer or carrier, or perform other operations that are otherwise inaccessible to the typical Android user. Rooting allows Android users to attain privileged control (known as "root access") within Android's subsystem. The rooting process involves exploiting security vulnerabilities in the device's firmware, copying the su binary to a location in the current process's PATH (e.g., /system/xbin/su), and granting it executable permissions with the chmod command.

Rooting enables all the user-installed applications to run privileged commands such as

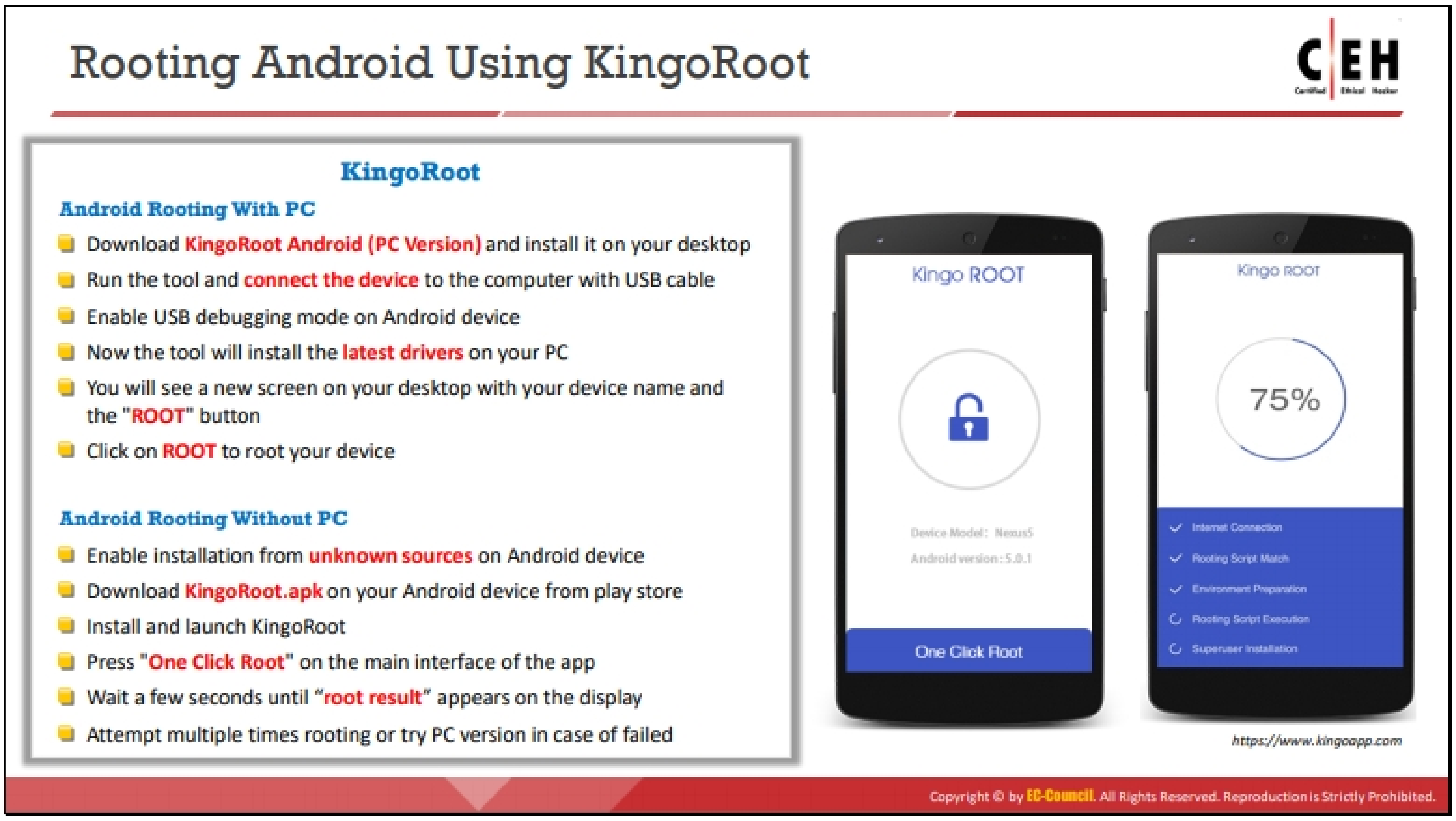
- Modifying or deleting system files, modules, ROMs (stock firmware), and kernels
- Removing carrier- or manufacturer-installed applications (bloatware)
- Low-level access to hardware that is typically unavailable to devices in their default configuration
- Improved performance
- Wi-Fi and Bluetooth tethering
- Installing applications on SD card
- Better user interface and keyboard

Rooting also comes with many security risks and other risks to your device, including

- Voiding your phone's warranty

- Poor performance
- Malware infection
- “Bricking” the device

One can use tools such as KingoRoot, TunesGo Root Android Tool, and so on to root Android devices.



Rooting Android Using KingoRoot

KingoRoot

Android Rooting With PC

- Download **KingoRoot Android (PC Version)** and install it on your desktop
- Run the tool and **connect the device** to the computer with USB cable
- Enable USB debugging mode on Android device
- Now the tool will install the **latest drivers** on your PC
- You will see a new screen on your desktop with your device name and the "**ROOT**" button
- Click on **ROOT** to root your device

Android Rooting Without PC

- Enable installation from **unknown sources** on Android device
- Download **KingoRoot.apk** on your Android device from play store
- Install and launch KingoRoot
- Press "**One Click Root**" on the main interface of the app
- Wait a few seconds until "**root result**" appears on the display
- Attempt multiple times rooting or try PC version in case of failed

https://www.kingoapp.com

Rooting Android Using KingoRoot

Source: *https://www.kingoapp.com*

KingoRoot is a tool used to root Android devices. It can be used with or without a PC. KingoRoot helps users root their Android devices to achieve the following:

- Preserve battery life
- Access root-only apps
- Remove carrier "bloatware"
- Customize appearance
- Attain admin level permission

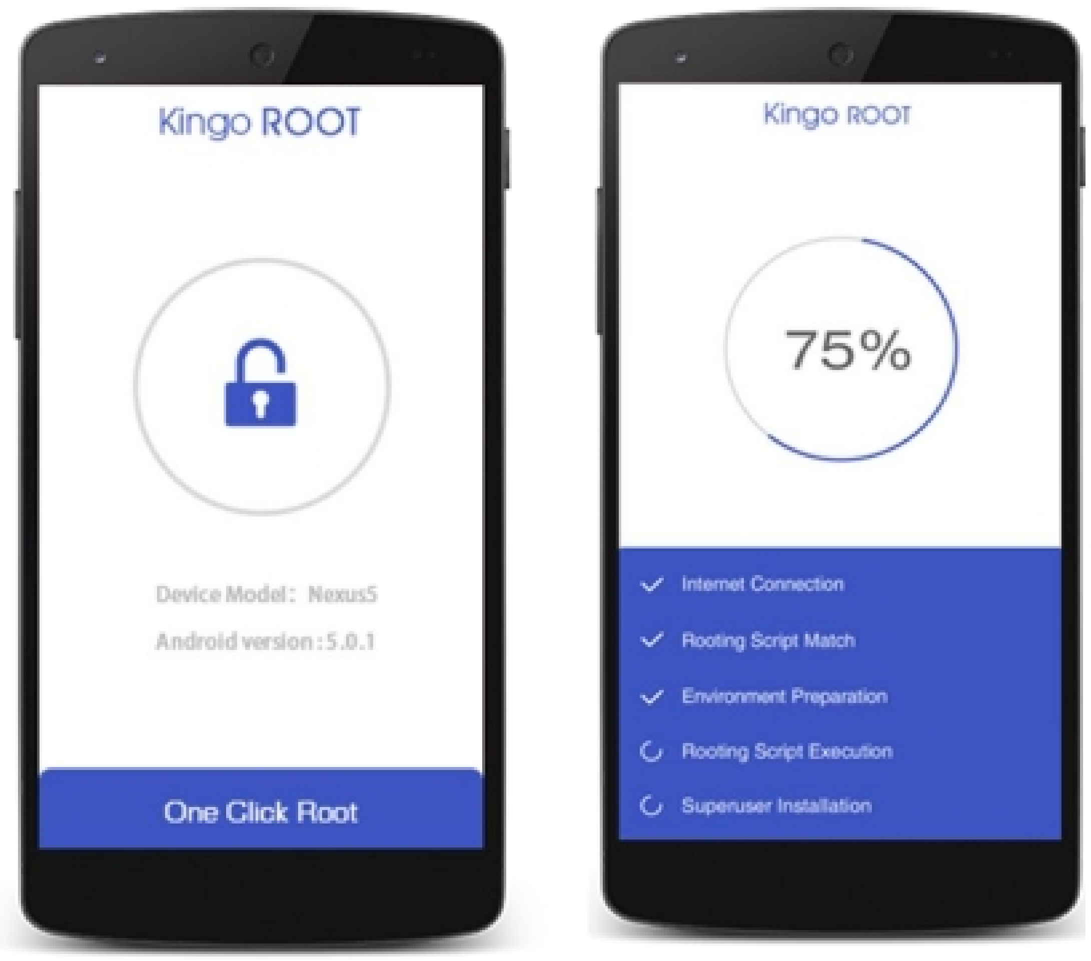
The following steps are involved in rooting an Android device with this tool:

Android Rooting With PC:

- Download KingoRoot Android (PC Version) and install it on your desktop.
- Run the tool and connect the device to the computer with a USB cable.
- Enable the USB debugging mode on your Android device.
- Now, the tool will install the latest drivers on your PC.
- You will see a new screen on your desktop with your device name and the "**ROOT**" button.
- Click on **ROOT** to root your device.

Android Rooting Without PC:

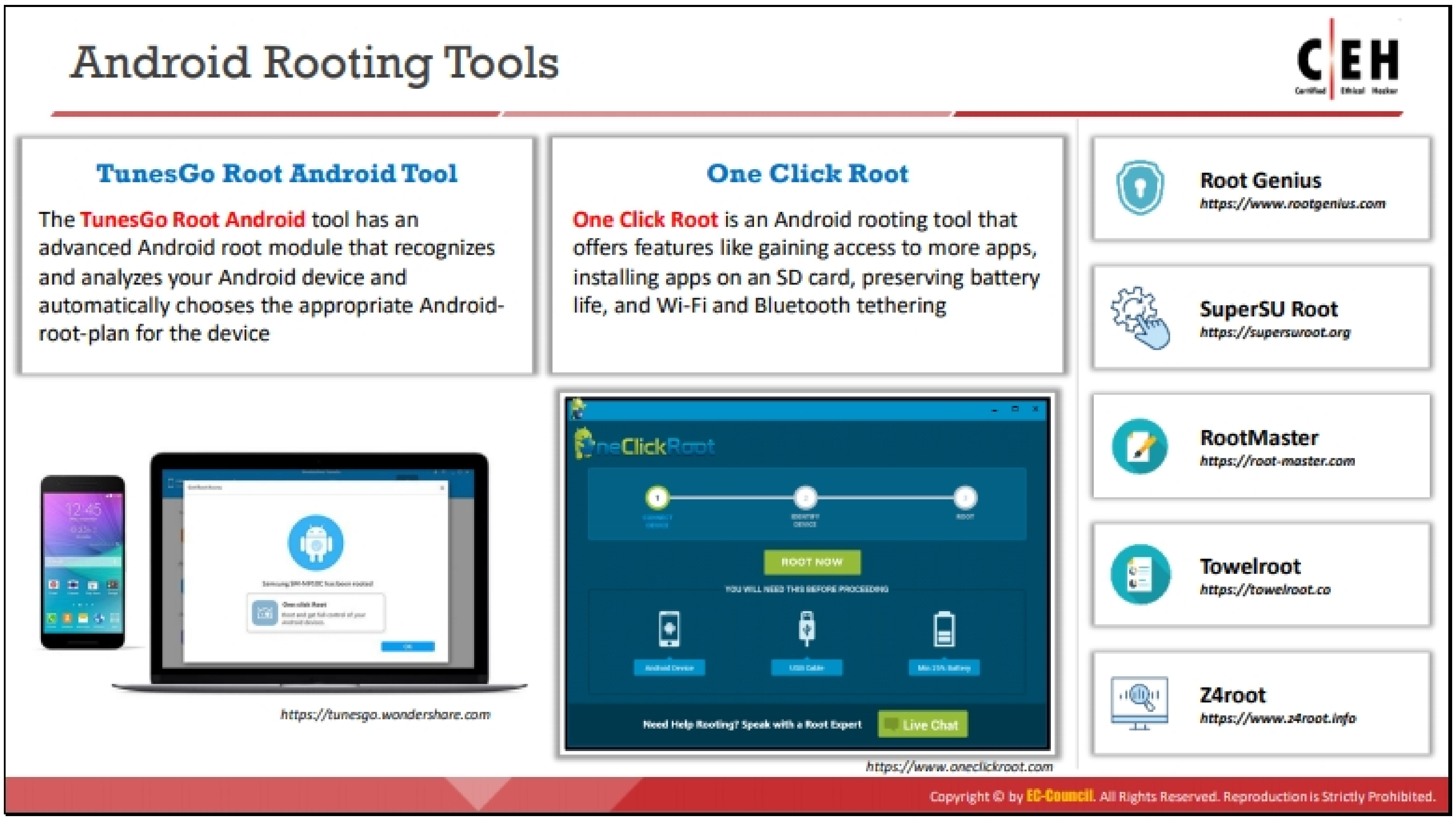
- Enable installation from unknown sources in your Android device.
- Download **KingoRoot.apk** on your Android device from Play Store.
- Install and launch KingoRoot.
- Press "**One Click Root**" on the main interface of the app.
- Wait for a few seconds until the root result appears on the display.
- Attempt multiple times in case of failed rooting or you can try the PC version.



Device Model: Nexus5
Android version: 5.0.1

✓ Internet Connection
✓ Rooting Script Match
✓ Environment Preparation
Rooting Script Execution
Superuser Installation

Figure 17.14: Screenshot of KingoRoot Android (APK Version)



Android Rooting Tools

TunesGo Root Android Tool

The **TunesGo Root Android** tool has an advanced Android root module that recognizes and analyzes your Android device and automatically chooses the appropriate Android-root-plan for the device

https://tunesgo.wondershare.com

One Click Root

One Click Root is an Android rooting tool that offers features like gaining access to more apps, installing apps on an SD card, preserving battery life, and Wi-Fi and Bluetooth tethering

OneClickRoot
ROOT NOW
YOU WILL NEED THIS BEFORE PROCEEDING
Need Help Rooting? Speak with a Root Expert
Live Chat

https://www.oneclickroot.com

Root Genius
https://www.rootgenius.com

SuperSU Root
https://supersuroot.org

RootMaster
https://root-master.com

Towelroot
https://towelroot.co

Z4root
https://www.z4root.info

Android Rooting Tools

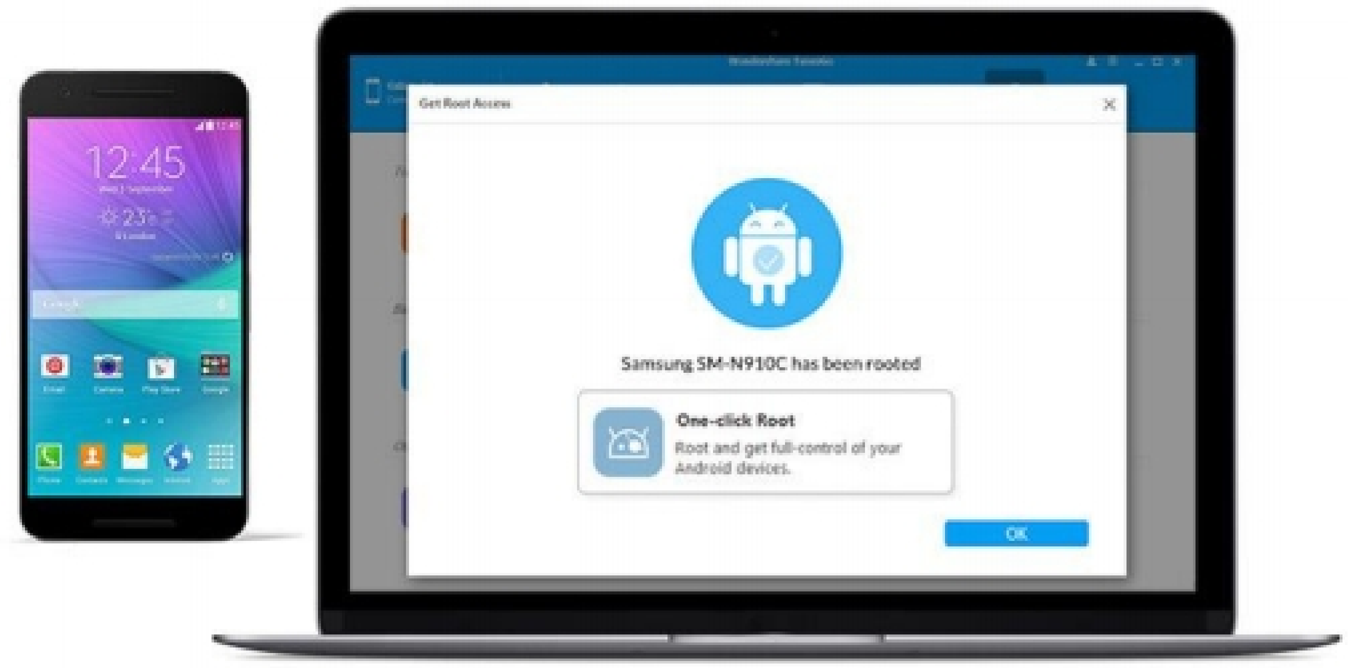
- **TunesGo Root Android Tool**

  Source: *https://tunesgo.wondershare.com*

  This tool has an advanced Android root module that recognizes and analyzes your Android device and automatically chooses the appropriate Android root plan for it.

  The steps to root an Android device using TunesGo Root Android tool are as follows:

  - Download the **TunesGo Root Android** tool
  - Connect your device to your computer
  - Find "**One-click Android Root**" in the toolbox and click on it to root your device
  - Your Android device is successfully rooted



Samsung SM-N910C has been rooted

One-click Root
Root and get full-control of your Android devices.

Figure 17.15: Screenshot of TunesGo Root Android

- **One Click Root**

  Source: *https://www.oneclickroot.com*

  One Click Root is an Android rooting software that supports most devices. It comes with extra fail-safes (such as instant unrooting) and offers full technical support. It allows rooting of an Android smartphone or tablet and provides access to additional features such as gaining access to more apps, installing apps on SD cards, preserving battery life, Wi-Fi and Bluetooth tethering, installing custom ROMs, and accessing blocked features.

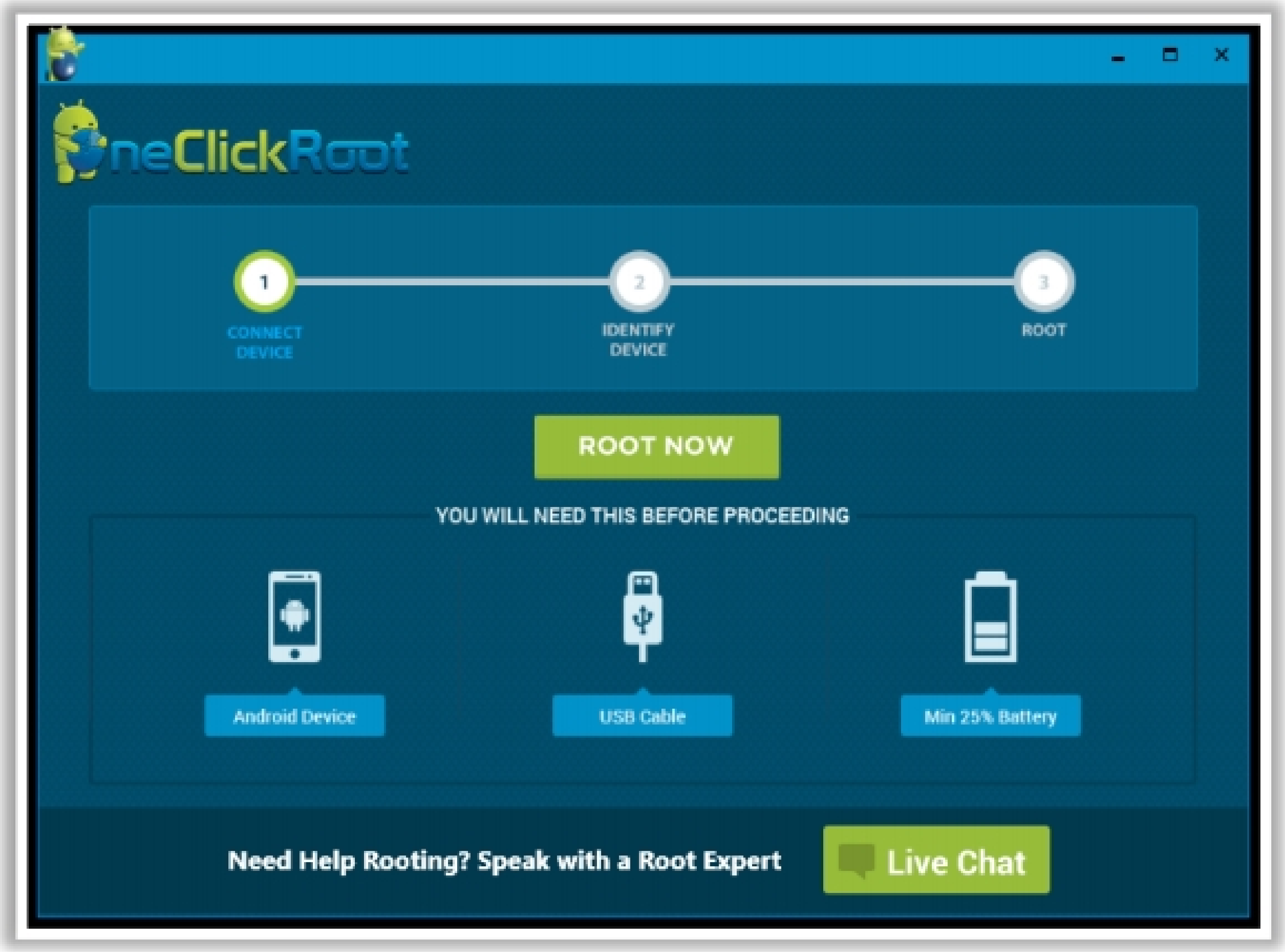
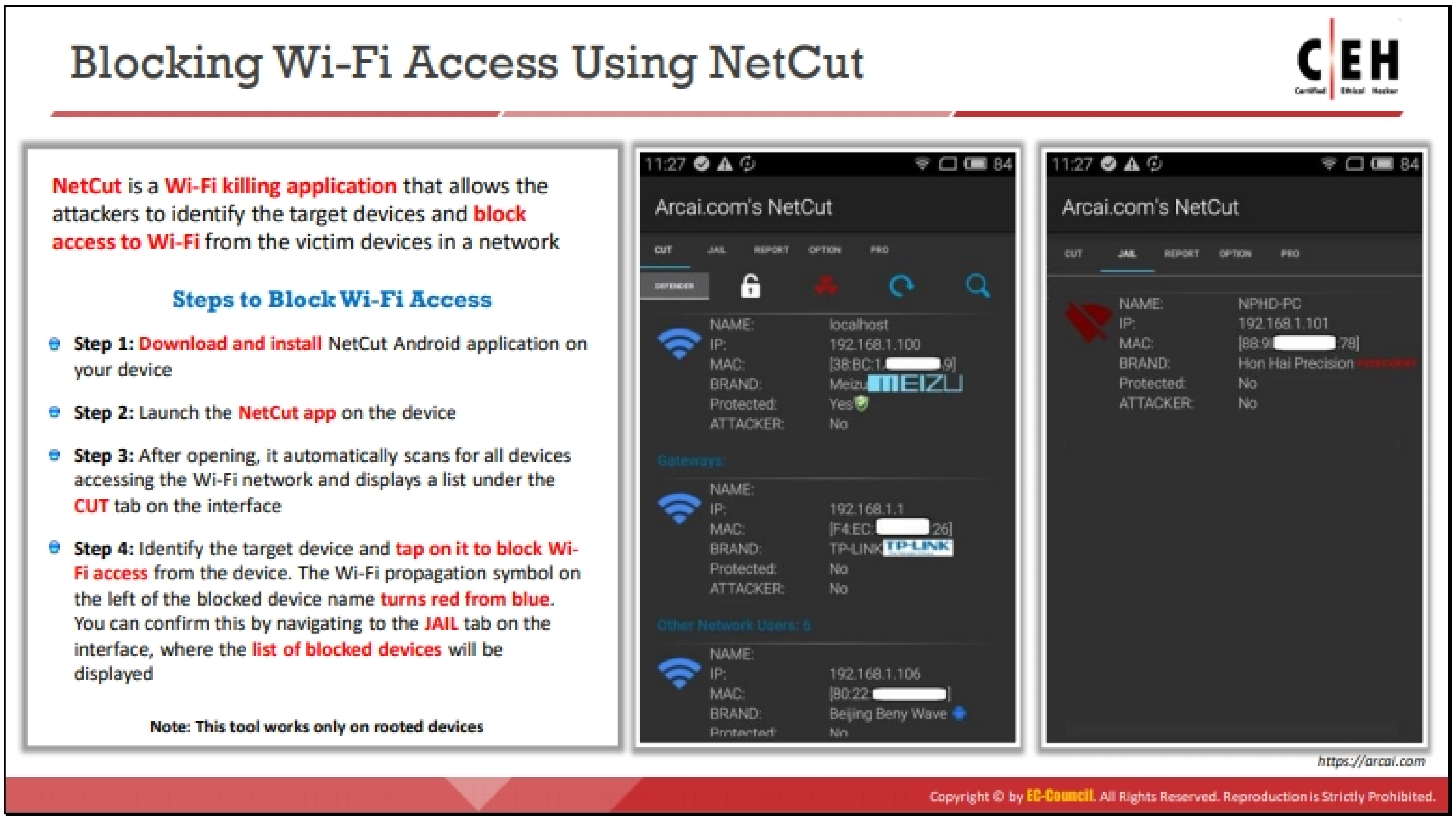

OneClickRoot
1
CONNECT DEVICE
2
IDENTIFY DEVICE
3
ROOT
ROOT NOW
YOU WILL NEED THIS BEFORE PROCEEDING
Android Device
USB Cable
Min 25% Battery
Need Help Rooting? Speak with a Root Expert
Live Chat

Figure 17.16: Screenshot of One Click Root

Some additional Android rooting tools are as follows:

- Root Genius (*https://www.rootgenius.com*)
- SuperSU Root (*https://supersuroot.org*)
- RootMaster (*https://root-master.com*)
- Towelroot (*https://towelroot.co*)
- Z4root (*https://www.z4root.info*)



Blocking Wi-Fi Access Using NetCut

NetCut is a **Wi-Fi killing application** that allows the attackers to identify the target devices and **block access to Wi-Fi** from the victim devices in a network

Steps to Block Wi-Fi Access

- **Step 1:** Download and install NetCut Android application on your device
- **Step 2:** Launch the NetCut app on the device
- **Step 3:** After opening, it automatically scans for all devices accessing the Wi-Fi network and displays a list under the CUT tab on the interface
- **Step 4:** Identify the target device and tap on it to block Wi-Fi access from the device. The Wi-Fi propagation symbol on the left of the blocked device name turns red from blue. You can confirm this by navigating to the JAIL tab on the interface, where the list of blocked devices will be displayed

Note: This tool works only on rooted devices

NAME: localhost
IP: 192.168.1.100
MAC: [38:BC:1[redacted]9]
BRAND: Meizu
Protected: Yes
ATTACKER: No

NAME:
IP: 192.168.1.1
MAC: [F4:EC:[redacted]26]
BRAND: TP-LINK
Protected: No
ATTACKER: No

NAME:
IP: 192.168.1.106
MAC: [80:22:[redacted]]
BRAND: Beijing Beny Wave
Protected: No

NAME: NPHD-PC
IP: 192.168.1.101
MAC: [88:9[redacted]:78]
BRAND: Hon Hai Precision
Protected: No
ATTACKER: No

https://arcai.com

Hacking Android Devices

Owing to the rapidly growing number of users of Android devices, these devices have become primary targets for most hackers. Attackers use various tools such as NetCut, drozer, zANTI, Network Spoofer, Low Orbit Ion Cannon (LOIC), DroidSheep, Orbot Proxy, and so on to launch attacks on Android devices.

Blocking Wi-Fi Access Using NetCut

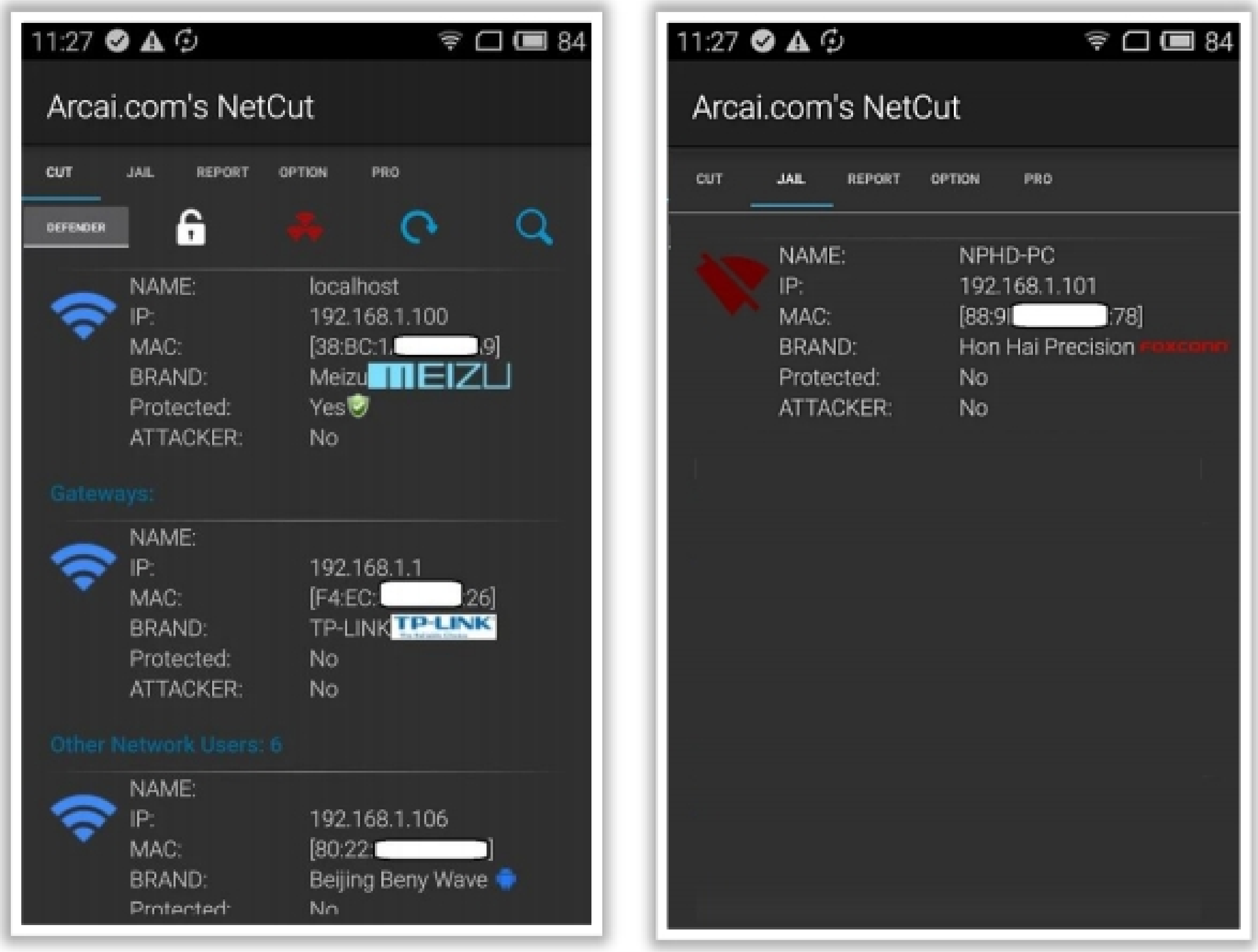
Source: *https://arcai.com*

NetCut is a Wi-Fi killing application that allows attackers in a network to identify target devices and block Wi-Fi access to these devices.

Note: This application works effectively only on rooted devices.

Follow the steps given below to block Wi-Fi access:

- **Step 1**: Download and install **NetCut** Android application on your device.
- **Step 2**: Launch the NetCut app.
- **Step 3**: It automatically scans all the devices accessing the Wi-Fi network and displays the list under the **CUT** tab on the interface.
- **Step 4**: Identify the target device and tap on it to block Wi-Fi access to the device. The Wi-Fi propagation symbol on the left of the blocked device name turns from blue to red. You can confirm this by navigating to the **JAIL** tab on the interface, where the list of blocked devices will be displayed.


11:27
84
Arcai.com's NetCut
CUT
JAIL
REPORT
OPTION
PRO
DEFENDER
MEIZU
TP-LINK
FOXCONN

NAME: localhost
IP: 192.168.1.100
MAC: [38:BC:1[redacted]9]
BRAND: Meizu
Protected: Yes
ATTACKER: No

Gateways:

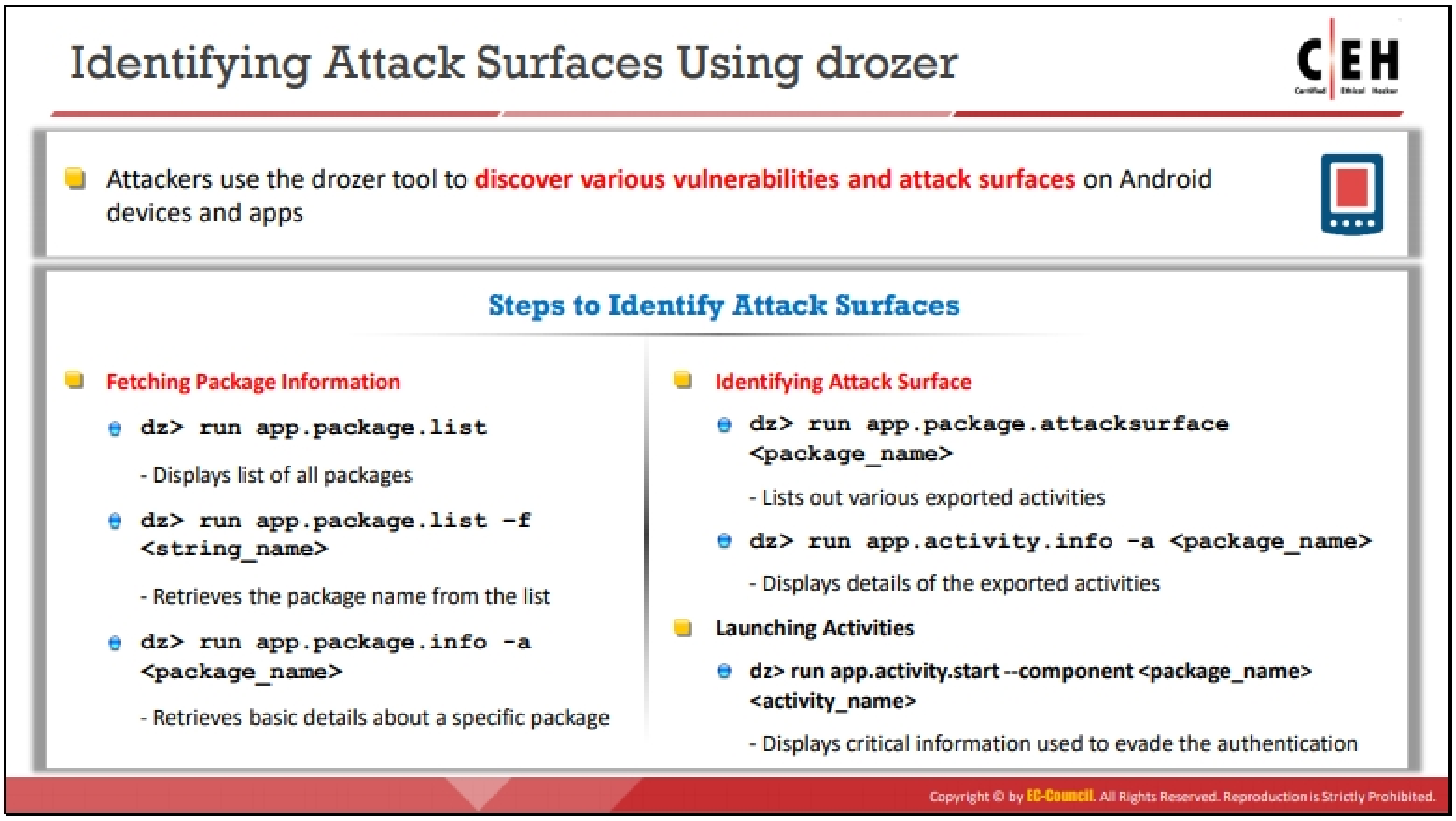
NAME:
IP: 192.168.1.1
MAC: [F4:EC:[redacted]26]
BRAND: TP-LINK
Protected: No
ATTACKER: No

Other Network Users: 6

NAME:
IP: 192.168.1.106
MAC: [80:22:[redacted]]
BRAND: Beijing Beny Wave
Protected: No

NAME: NPHD-PC
IP: 192.168.1.101
MAC: [88:9[redacted]:78]
BRAND: Hon Hai Precision
Protected: No
ATTACKER: No

Figure 17.17: Screenshot of NetCut



Identifying Attack Surfaces Using drozer

- Attackers use the drozer tool to **discover various vulnerabilities and attack surfaces** on Android devices and apps

Steps to Identify Attack Surfaces

- **Fetching Package Information**
  - `dz> run app.package.list`
    - Displays list of all packages
  - `dz> run app.package.list -f <string_name>`
    - Retrieves the package name from the list
  - `dz> run app.package.info -a <package_name>`
    - Retrieves basic details about a specific package
- **Identifying Attack Surface**
  - `dz> run app.package.attacksurface <package_name>`
    - Lists out various exported activities
  - `dz> run app.activity.info -a <package_name>`
    - Displays details of the exported activities
- **Launching Activities**
  - **`dz> run app.activity.start --component <package_name> <activity_name>`**
    - Displays critical information used to evade the authentication

Identifying Attack Surfaces Using drozer

Attackers use the drozer tool to discover various vulnerabilities and attack surfaces on Android devices and apps. It is also incorporated with various features to control remote Android devices. Attackers do not need any USB debugging techniques; they can assess the device using drozer in the production state itself. This tool offers a drozer agent (emulator used for testing) and drozer console (command-line interface) through its package that can be leveraged by an attacker to perform various assessment operations on target devices.

After installing the drozer agent, the steps given below must be followed to identify attack surfaces on the target Android device:

- **Fetching Package Information**

  Use the following commands to fetch the package information from a connected device:

  ```
  dz> run app.package.list
  ```

  - Displays all the packages inside the device

  ```
  dz> run app.package.list -f <string_name>
  ```

  - Retrieves the package name from the list

  ```
  dz> run app.package.info -a <package_name>
  ```

  - Retrieves basic details about a specific package

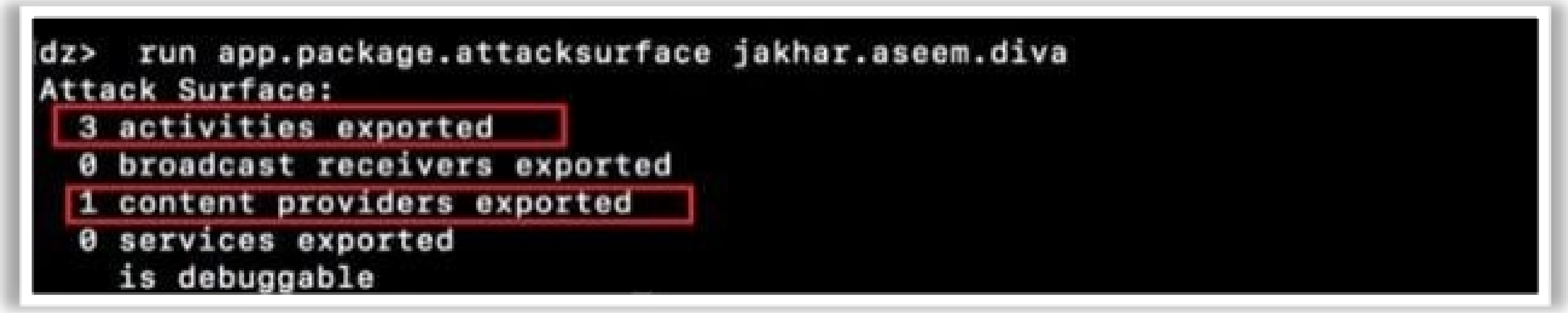
  By running the abovementioned commands, an attacker obtains all the information about the required package.

- **Identifying Attack Surface**

  Now, the attacker uses utilities from the abovementioned package to identify attack surfaces on the device. Use the following commands to list information on exported activities, services, broadcast receivers, and content providers:

  `dz> run app.package.attacksurface <package_name>`

  - Lists various exported activities

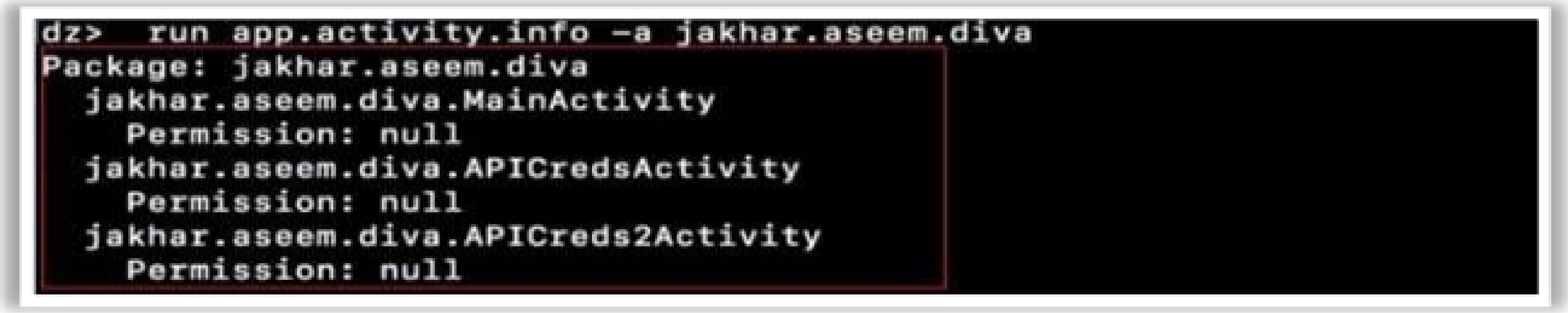


```
dz>  run app.package.attacksurface jakhar.aseem.diva
Attack Surface:
  3 activities exported
  0 broadcast receivers exported
  1 content providers exported
  0 services exported
    is debuggable
```

Figure 17.18: Screenshot of Dozer identifying attack surface

  `dz> run app.activity.info -a <package_name>`

  - Displays details of the exported activities



```
dz>  run app.activity.info -a jakhar.aseem.diva
Package: jakhar.aseem.diva
  jakhar.aseem.diva.MainActivity
    Permission: null
  jakhar.aseem.diva.APICredsActivity
    Permission: null
  jakhar.aseem.diva.APICreds2Activity
    Permission: null
```

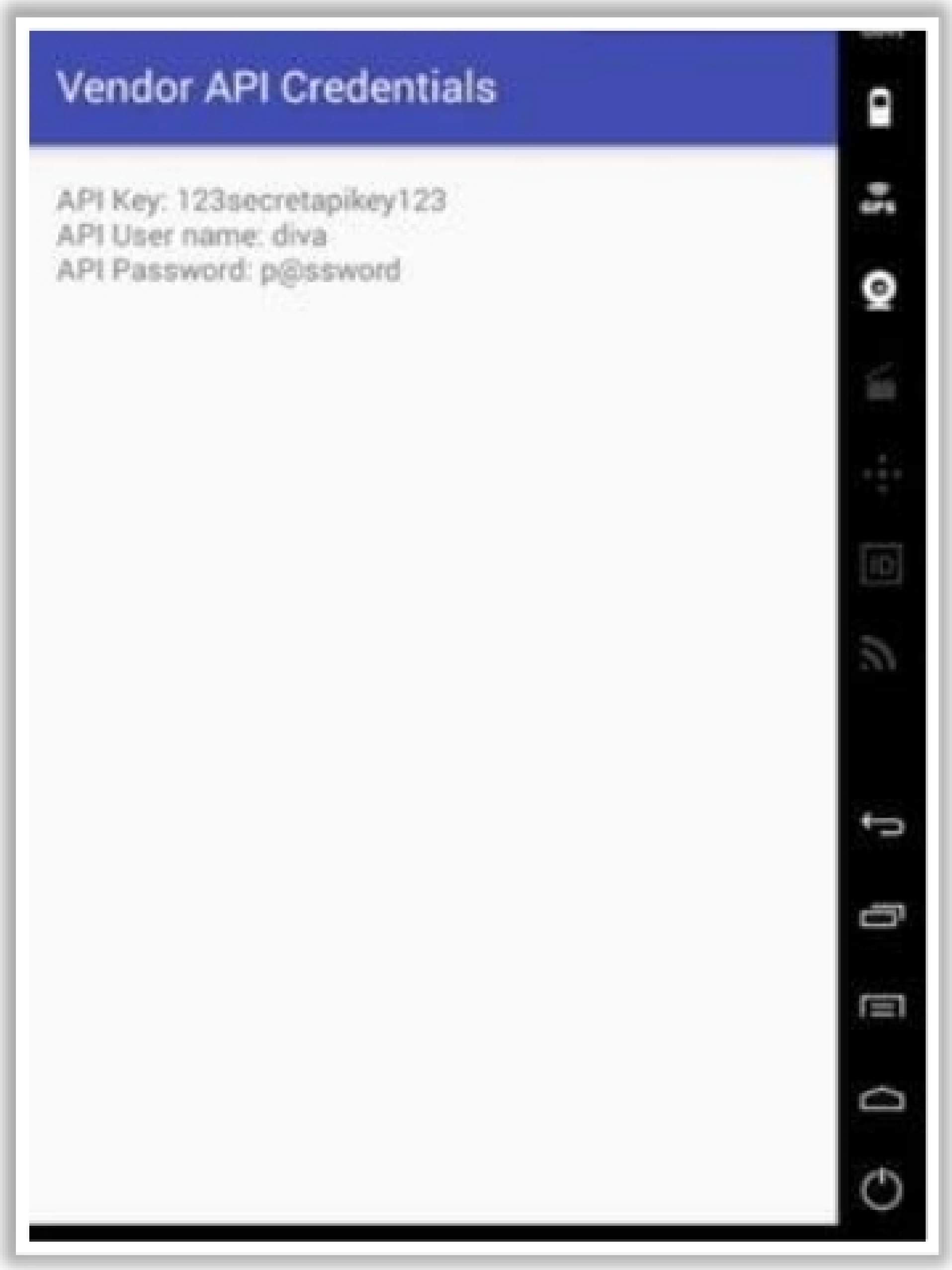
Figure 17.19: Screenshot of Dozer displaying activity information

- **Launching Activities**

  Use the following command to launch the required activity:

  `dz> run app.activity.start --component <package_name> <activity_name>`

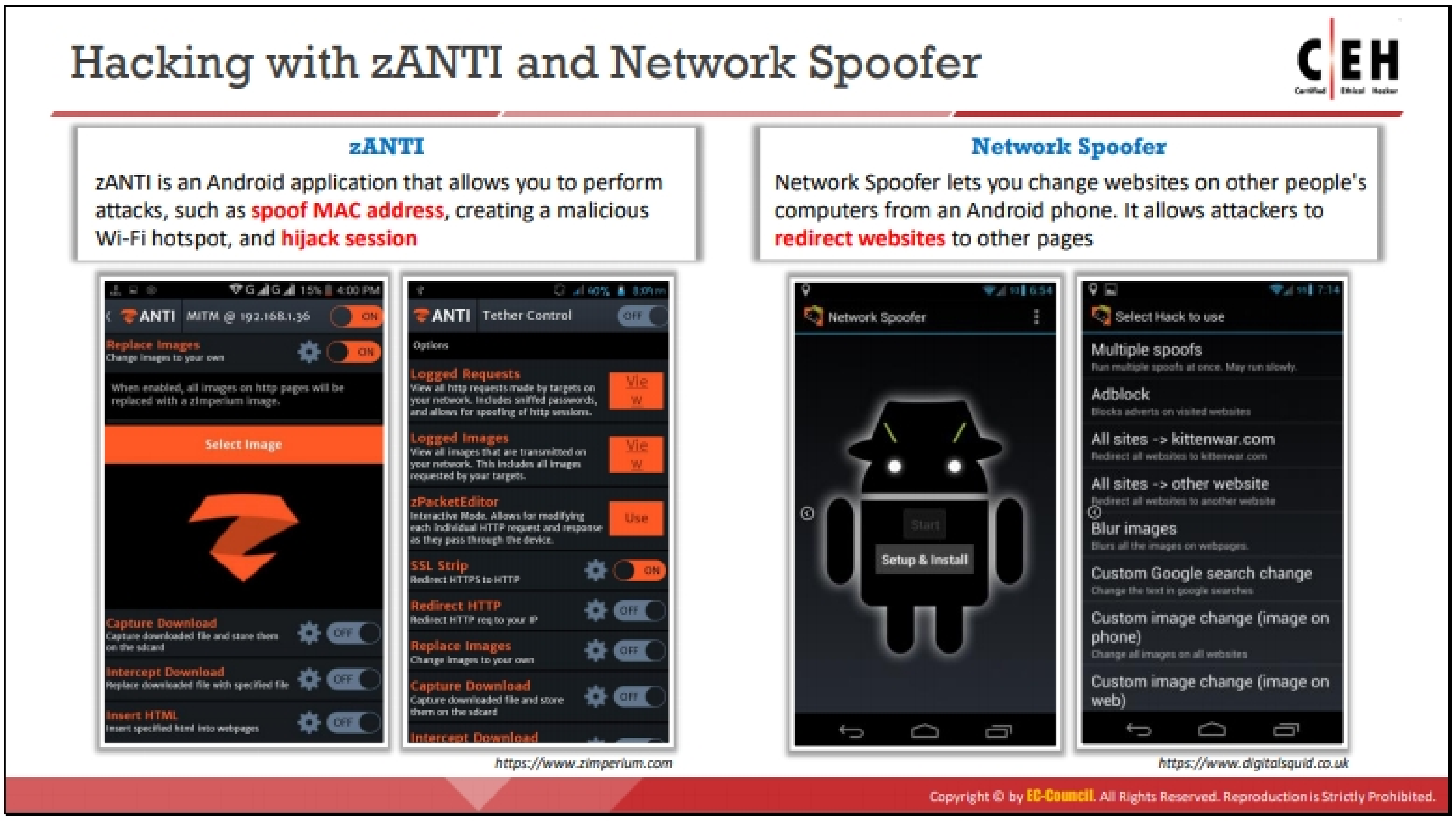
  The activity displays critical information that can be exploited to evade the authentication process.



API Key: 123secretapikey123
API User name: diva
API Password: p@ssword

Figure 17.20: Screenshot of Dozer displaying credentials

After bypassing the authentication process, the attacker can discover various attack surfaces and further exploit them to launch various attacks on target Android devices.



Hacking with zANTI and Network Spoofer

zANTI

zANTI is an Android application that allows you to perform attacks, such as **spoof MAC address**, creating a malicious Wi-Fi hotspot, and **hijack session**

Network Spoofer

Network Spoofer lets you change websites on other people's computers from an Android phone. It allows attackers to **redirect websites** to other pages

ANTI MITM @ 192.168.1.36 ON

Replace Images — Change images to your own — ON

When enabled, all images on http pages will be replaced with a zimperium image.

Select Image

Capture Download — Capture downloaded file and store them on the sdcard — OFF

Intercept Download — Replace downloaded file with specified file — OFF

Insert HTML — Insert specified html into webpages — OFF

ANTI Tether Control OFF

Options

Logged Requests — View all http requests made by targets on your network. Includes sniffed passwords, and allows for spoofing of http sessions. — View

Logged Images — View all images that are transmitted on your network. This includes all images requested by your targets. — View

zPacketEditor — Interactive Mode. Allows for modifying each individual HTTP request and response as they pass through the device. — Use

SSL Strip — Redirect HTTPS to HTTP — ON

Redirect HTTP — Redirect HTTP req to your IP — OFF

Replace Images — Change images to your own — OFF

Capture Download — Capture downloaded file and store them on the sdcard — OFF

Intercept Download

https://www.zimperium.com

Network Spoofer

Start

Setup & Install

Select Hack to use

Multiple spoofs — Run multiple spoofs at once. May run slowly.

Adblock — Blocks adverts on visited websites

All sites -> kittenwar.com — Redirect all websites to kittenwar.com

All sites -> other website — Redirect all websites to another website

Blur images — Blurs all the images on webpages.

Custom Google search change — Change the text in google searches

Custom image change (image on phone) — Change all images on all websites

Custom image change (image on web)

https://www.digitalsquid.co.uk

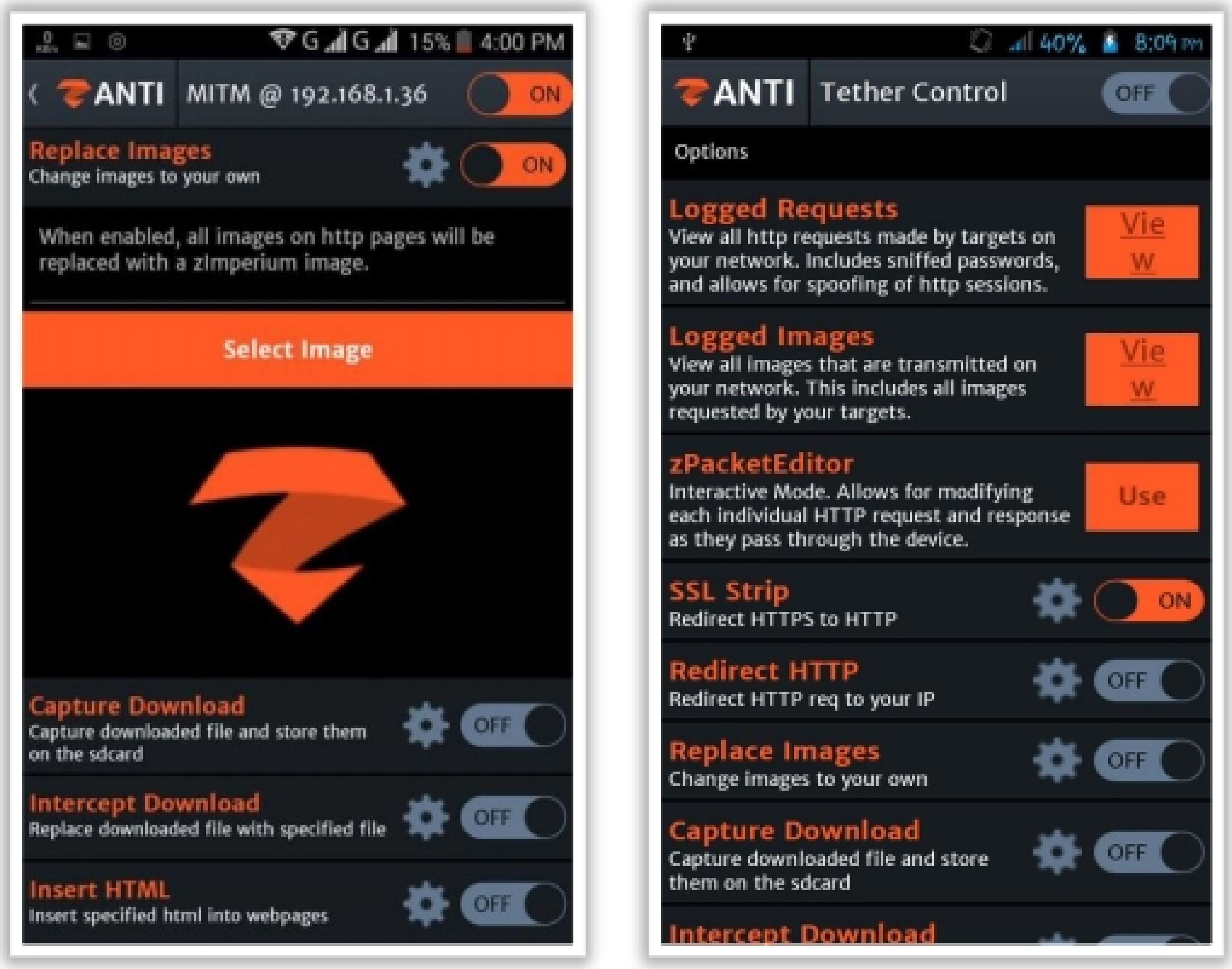
Hacking with zANTI and Network Spoofer

- **Hacking Networks Using zANTI**

  Source: *https://www.zimperium.com*

  zANTI is an Android application that allows you to perform the following attacks:

  - Spoof MAC Address
  - Create malicious Wi-Fi hotspot to capture victims to control and hijack their device traffic
  - Scan for open ports
  - Exploit router vulnerabilities
  - Password complexity audits
  - MITM and DoS attack
  - View, modify, and redirect all HTTP requests and responses
  - Redirect HTTPS to HTTP; redirect HTTP request to a particular IP or web page
  - Insert HTML code into web pages
  - Hijack sessions
  - View and replace all images that are transmitted over the network
  - Capture and intercept downloads



Replace Images – Change images to your own – ON

When enabled, all images on http pages will be replaced with a zImperium image.

Capture Download – Capture downloaded file and store them on the sdcard – OFF

Intercept Download – Replace downloaded file with specified file – OFF

Insert HTML – Insert specified html into webpages – OFF

Logged Requests – View all http requests made by targets on your network. Includes sniffed passwords, and allows for spoofing of http sessions.

Logged Images – View all images that are transmitted on your network. This includes all images requested by your targets.

zPacketEditor – Interactive Mode. Allows for modifying each individual HTTP request and response as they pass through the device.

SSL Strip – Redirect HTTPS to HTTP – ON

Redirect HTTP – Redirect HTTP req to your IP – OFF

Replace Images – Change images to your own – OFF

Capture Download – Capture downloaded file and store them on the sdcard – OFF

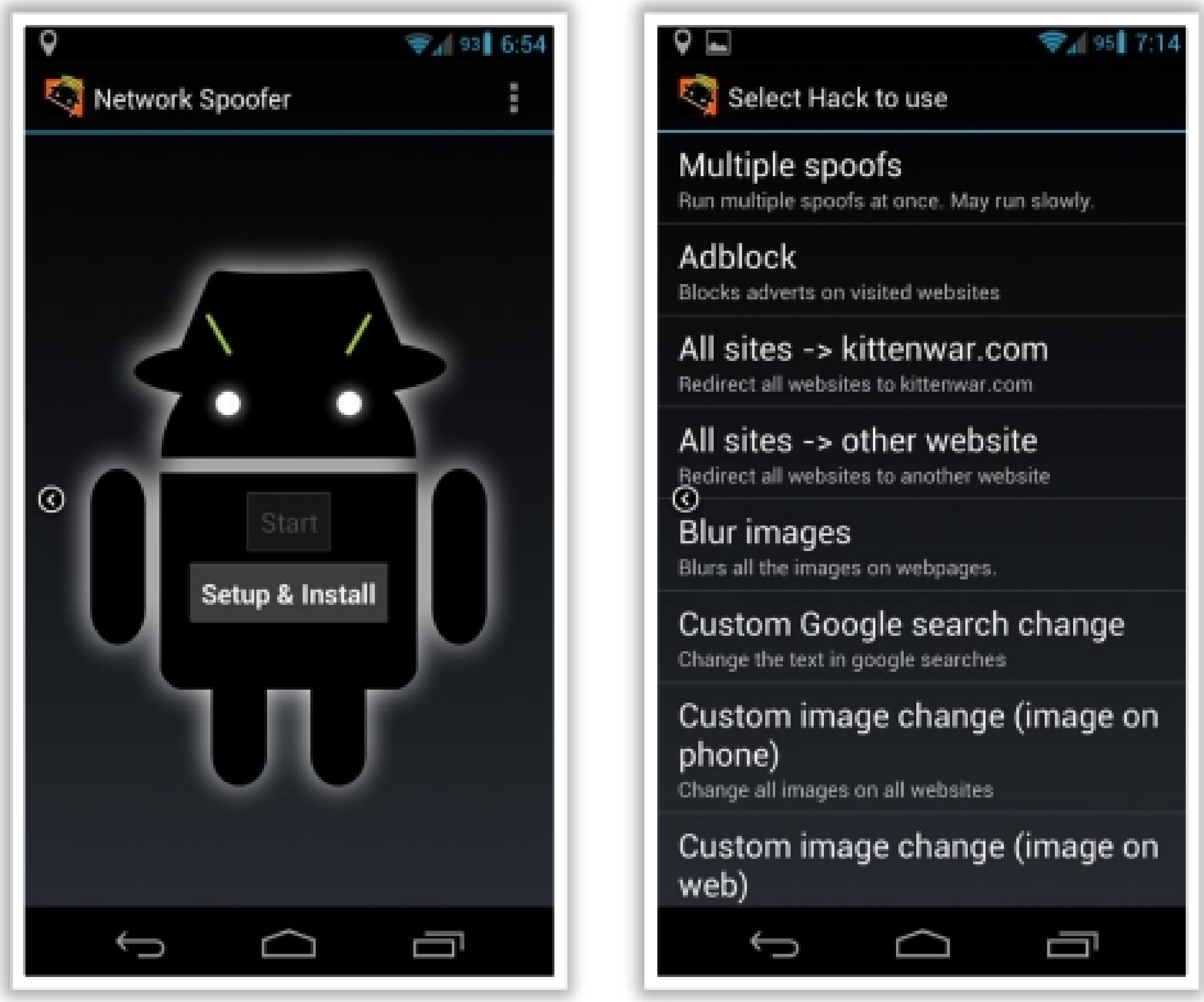
Intercept Download

Figure 17.21: Screenshot of zANTI

- **Hacking Networks Using Network Spoofer**

  Source: *https://www.digitalsquid.co.uk*

  Network Spoofer allows you to change websites on others' computers via an Android phone. It allows attackers to flip pictures and text upside down, make websites experience gravity, redirect websites to other pages, and delete or replace random words on websites.



Multiple spoofs – Run multiple spoofs at once. May run slowly.

Adblock – Blocks adverts on visited websites

All sites -> kittenwar.com – Redirect all websites to kittenwar.com

All sites -> other website – Redirect all websites to another website

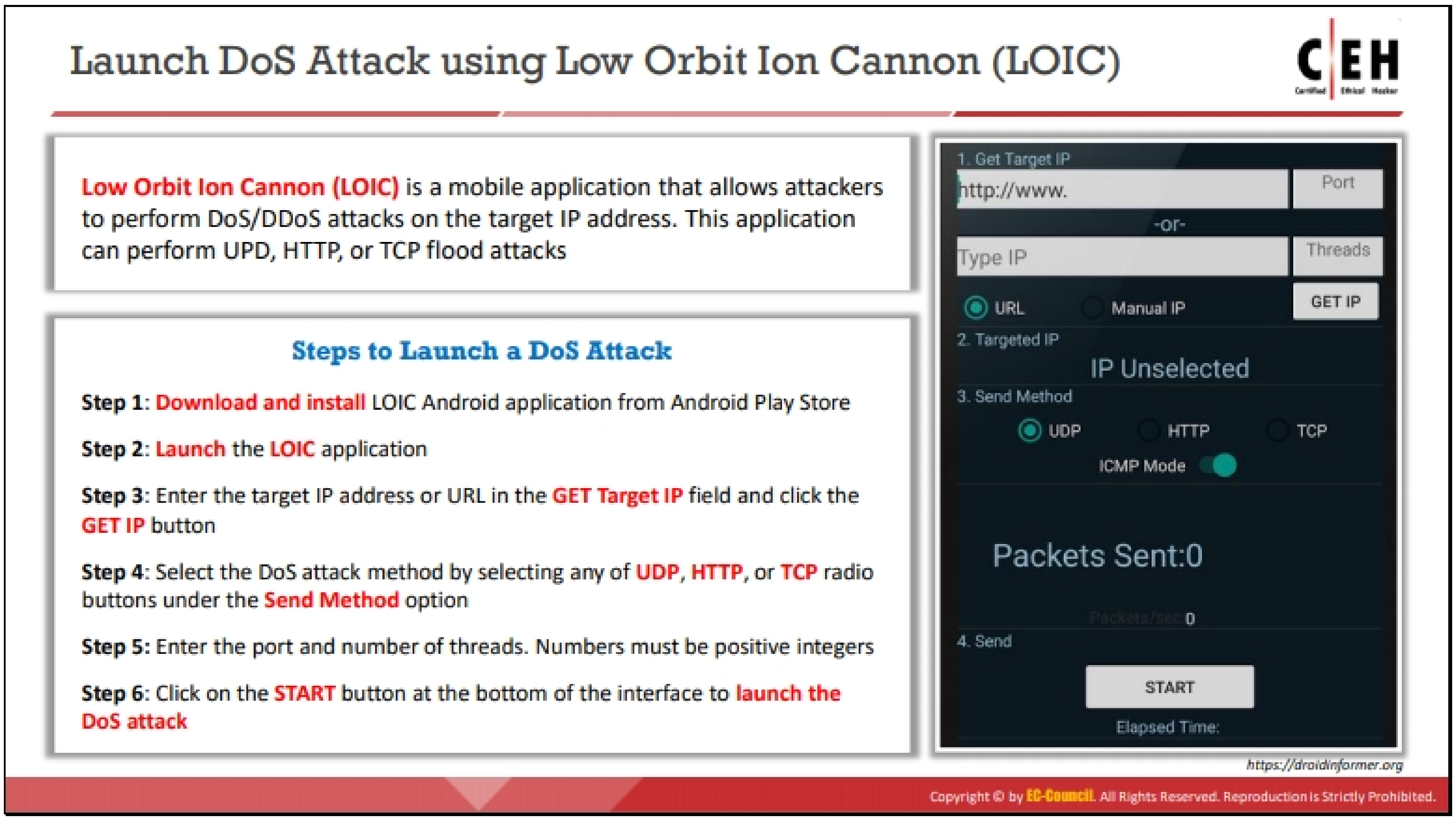
Blur images – Blurs all the images on webpages.

Custom Google search change – Change the text in google searches

Custom image change (image on phone) – Change all images on all websites

Custom image change (image on web)

Figure 17.22: Screenshot of Network Spoofer



Launch DoS Attack using Low Orbit Ion Cannon (LOIC)

Low Orbit Ion Cannon (LOIC) is a mobile application that allows attackers to perform DoS/DDoS attacks on the target IP address. This application can perform UPD, HTTP, or TCP flood attacks

Steps to Launch a DoS Attack

Step 1: **Download and install** LOIC Android application from Android Play Store

Step 2: **Launch** the **LOIC** application

Step 3: Enter the target IP address or URL in the **GET Target IP** field and click the **GET IP** button

Step 4: Select the DoS attack method by selecting any of **UDP**, **HTTP**, or **TCP** radio buttons under the **Send Method** option

Step 5: Enter the port and number of threads. Numbers must be positive integers

Step 6: Click on the **START** button at the bottom of the interface to **launch the DoS attack**

https://droidinformer.org

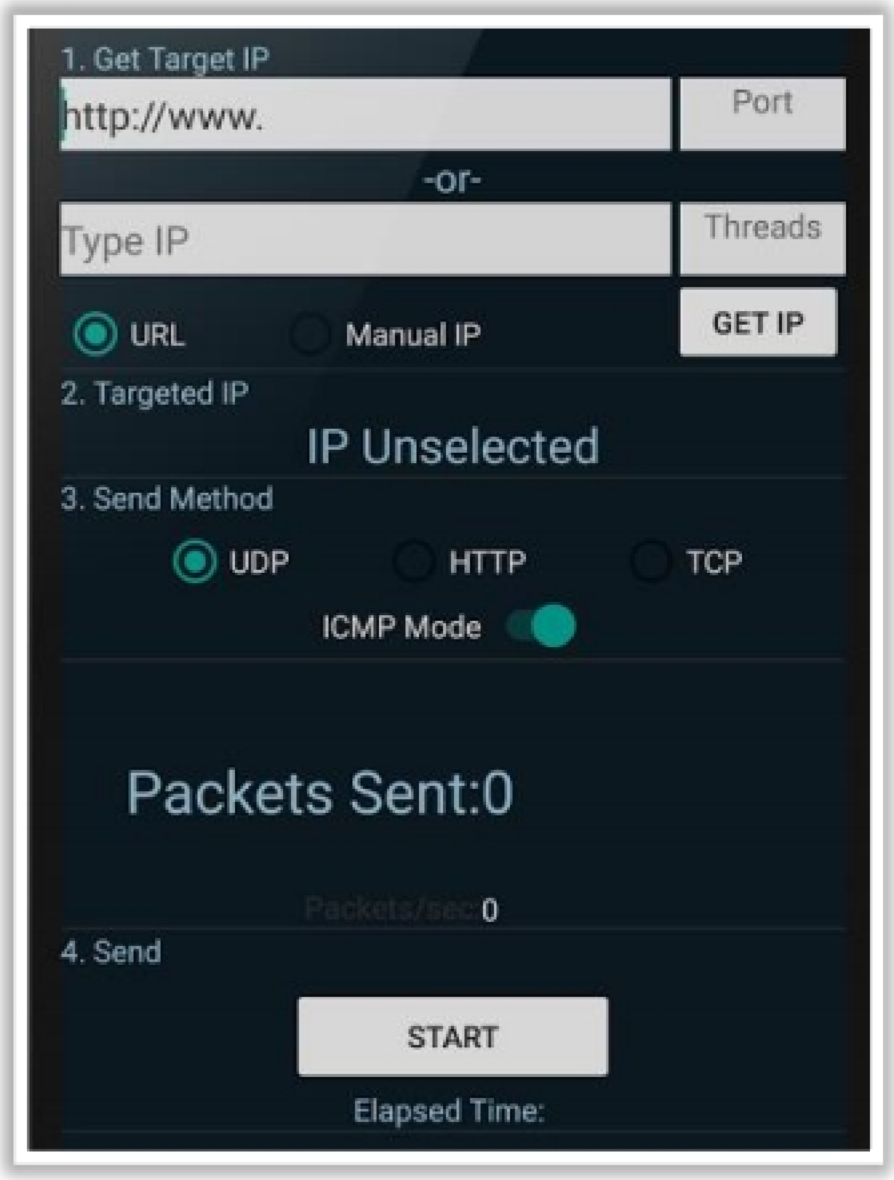
Launch DoS Attack using Low Orbit Ion Cannon (LOIC)

Source: *https://droidinformer.org*

LOIC is a mobile application that allows attackers to perform DoS/DDoS attacks on the target IP address. This application can perform UPD, HTTP, or TCP flood attacks. It allows attackers to take full control of the traffic flow, send data packets to any IP address, use various methods to send data packets (HTTP, UDP, or TCP), retrieve the IP address from any real web-address, and send data packets to any port.

Follow the steps given below to launch a DoS attack:

- **Step 1**: Download and install **the** LOIC Android application from Android Play Store.
- **Step 2**: Launch the LOIC application.
- **Step 3**: Enter the target IP address or the URL in the **GET Target IP** field and click **GET IP** button.
- **Step 4**: Select the DoS attack method by selecting any of the **UDP**, **HTTP**, or **TCP** radio buttons under the **Send Method** option.
- **Step 5:** Enter the port and number of threads. The numbers must be a positive whole number.
- **Step 6**: Click on the **START** button at the bottom of the interface to launch the DoS attack.


1. Get Target IP
http://www.
Port
-or-
Type IP
Threads
GET IP
3. Send Method
4. Send
START
Elapsed Time:

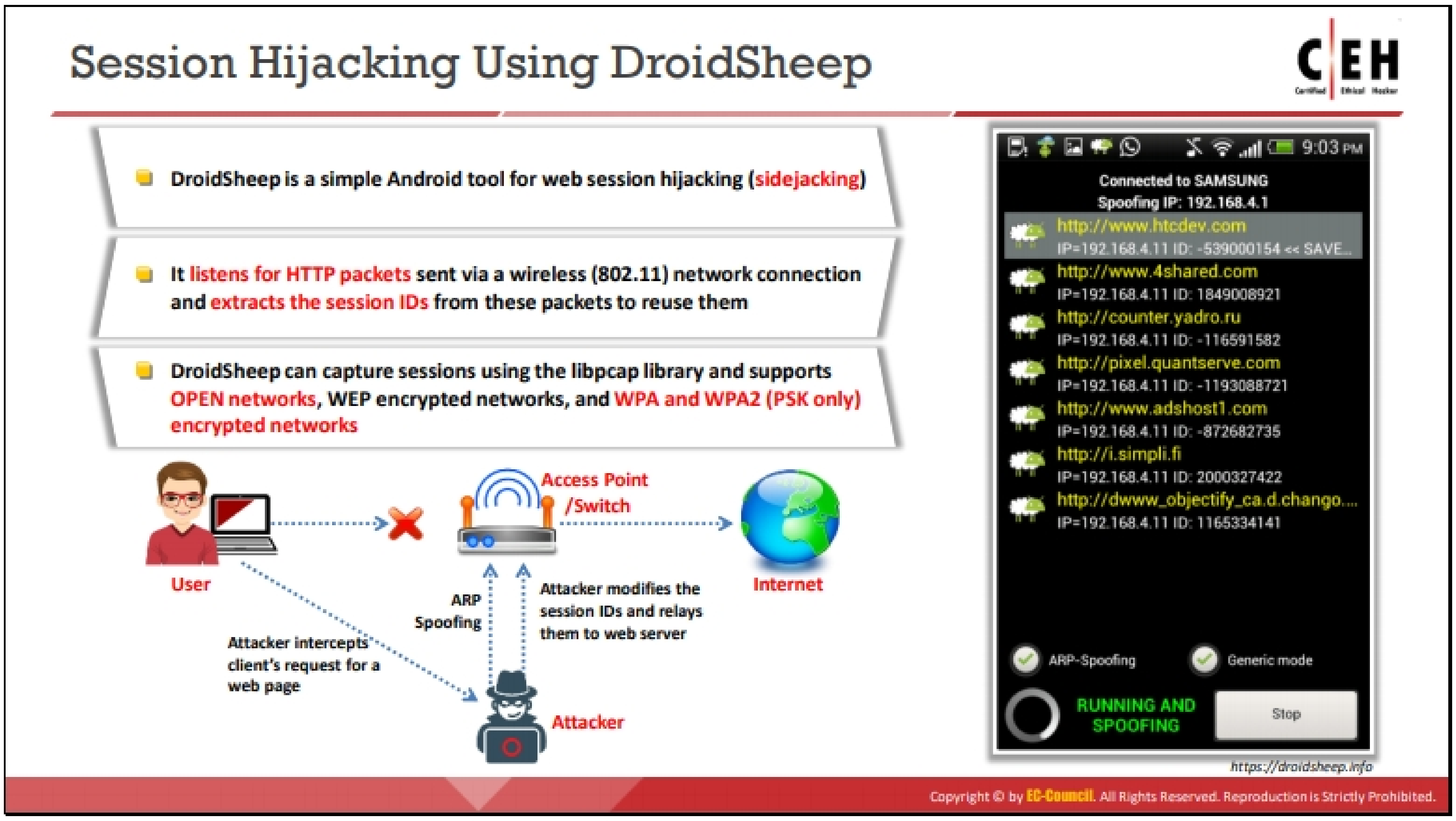
● URL ○ Manual IP

2. Targeted IP
IP Unselected

● UDP ○ HTTP ○ TCP
ICMP Mode: on

Packets Sent:0
Packets/sec: 0

Figure 17.23: Screenshot of Low Orbit Ion Cannon (LOIC)



Session Hijacking Using DroidSheep

- DroidSheep is a simple Android tool for web session hijacking (sidejacking)
- It listens for HTTP packets sent via a wireless (802.11) network connection and extracts the session IDs from these packets to reuse them
- DroidSheep can capture sessions using the libpcap library and supports OPEN networks, WEP encrypted networks, and WPA and WPA2 (PSK only) encrypted networks

Access Point /Switch
User
Internet
ARP Spoofing
Attacker modifies the session IDs and relays them to web server
Attacker intercepts client's request for a web page
Attacker

```
9:03 PM
Connected to SAMSUNG
Spoofing IP: 192.168.4.1
http://www.htcdev.com
IP=192.168.4.11 ID: -539000154 << SAVE...
http://www.4shared.com
IP=192.168.4.11 ID: 1849008921
http://counter.yadro.ru
IP=192.168.4.11 ID: -116591582
http://pixel.quantserve.com
IP=192.168.4.11 ID: -1193088721
http://www.adshost1.com
IP=192.168.4.11 ID: -872682735
http://i.simpli.fi
IP=192.168.4.11 ID: 2000327422
http://dwww_objectify_ca.d.chango....
IP=192.168.4.11 ID: 1165334141
```

ARP-Spoofing
Generic mode
RUNNING AND SPOOFING
Stop

https://droidsheep.info

Session Hijacking Using DroidSheep

Source: *https://droidsheep.info*

DroidSheep is a simple Android tool for web session hijacking ("**sidejacking**") using libpcap and arpspoof. Most web applications use a session ID to verify the user identity in the application. They transmit this session ID in subsequent requests in HTTP packets to maintain the user session. DroidSheep listens for HTTP packets sent via a wireless (802.11) network connection and extracts the session IDs from these packets to reuse them. Attackers can use DroidSheep to read all packets sent via a wireless network and capture the session ID. Once captured, the stolen session ID is used by the attacker to access the target web app on behalf of the victim. DroidSheep can capture sessions using the libpcap library and it supports OPEN networks, WEP encrypted networks, and WPA and WPA2 (PSK only) encrypted networks.

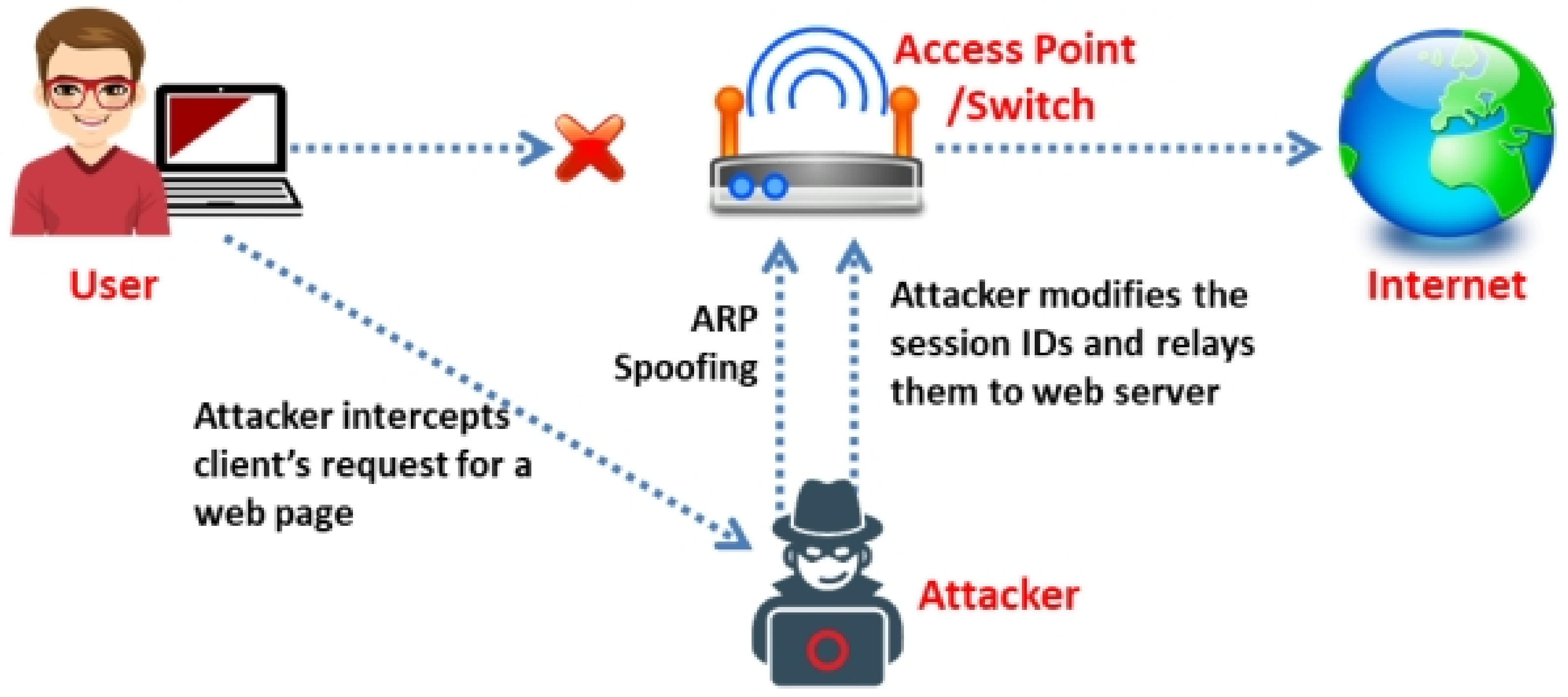
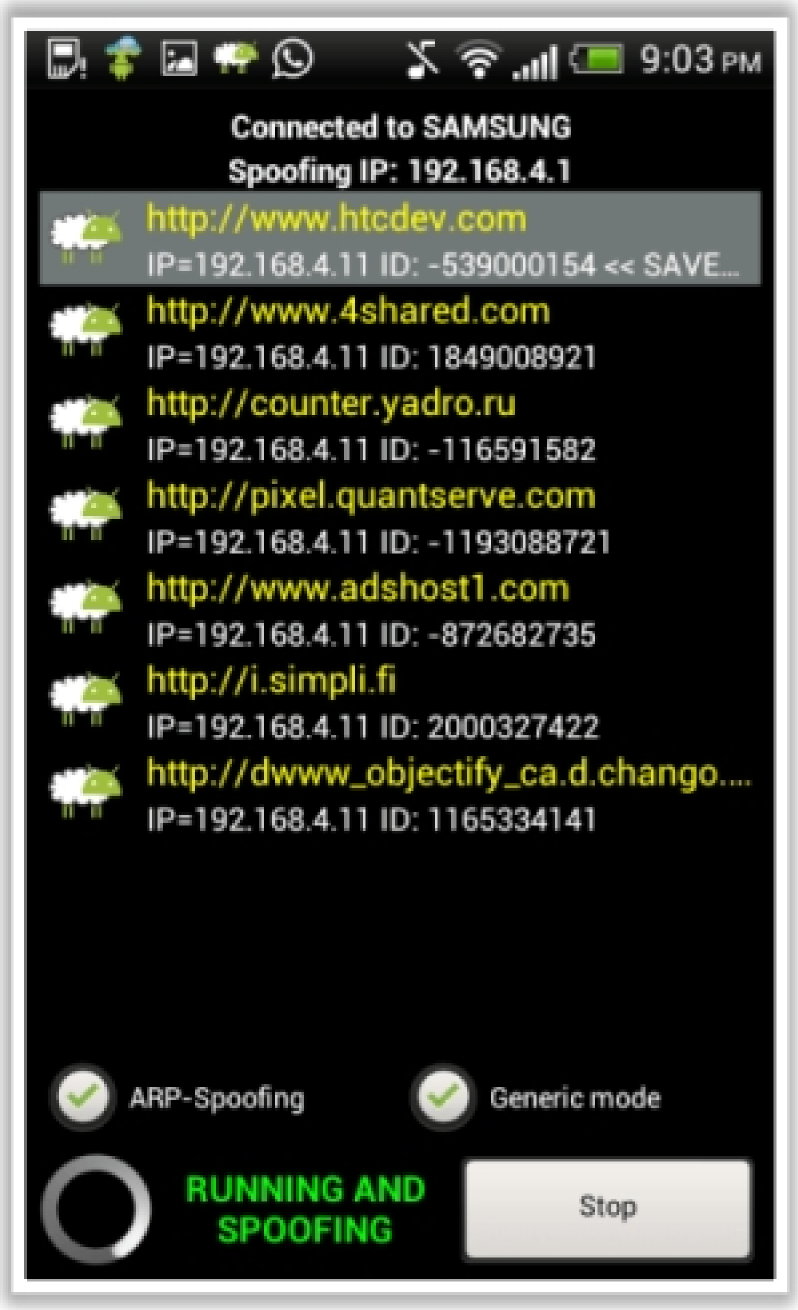

Access Point /Switch
User
Internet
ARP Spoofing
Attacker modifies the session IDs and relays them to web server
Attacker intercepts client's request for a web page
Attacker

Figure 17.24: Example of Session Hijacking attack with DroidSheep



Connected to SAMSUNG
Spoofing IP: 192.168.4.1

http://www.htcdev.com
IP=192.168.4.11 ID: -539000154 << SAVE...

http://www.4shared.com
IP=192.168.4.11 ID: 1849008921

http://counter.yadro.ru
IP=192.168.4.11 ID: -116591582

http://pixel.quantserve.com
IP=192.168.4.11 ID: -1193088721

http://www.adshost1.com
IP=192.168.4.11 ID: -872682735

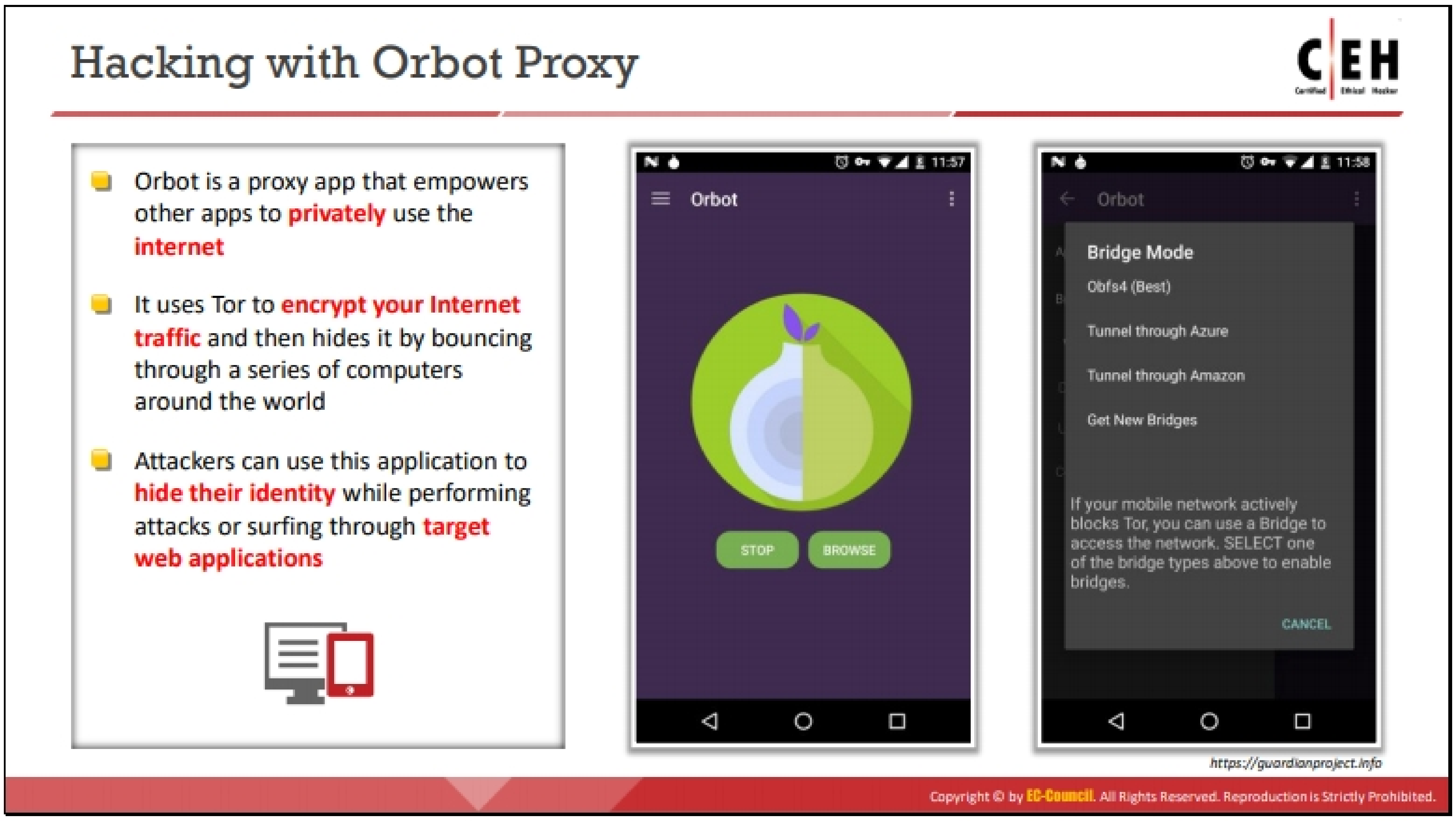
http://i.simpli.fi
IP=192.168.4.11 ID: 2000327422

http://dwww_objectify_ca.d.chango....
IP=192.168.4.11 ID: 1165334141

☑ ARP-Spoofing ☑ Generic mode

RUNNING AND SPOOFING

Figure 17.25: Screenshot of DroidSheep



Hacking with Orbot Proxy

- Orbot is a proxy app that empowers other apps to **privately** use the **internet**
- It uses Tor to **encrypt your Internet traffic** and then hides it by bouncing through a series of computers around the world
- Attackers can use this application to **hide their identity** while performing attacks or surfing through **target web applications**

Orbot
STOP
BROWSE
Orbot
Bridge Mode
Obfs4 (Best)
Tunnel through Azure
Tunnel through Amazon
Get New Bridges
CANCEL

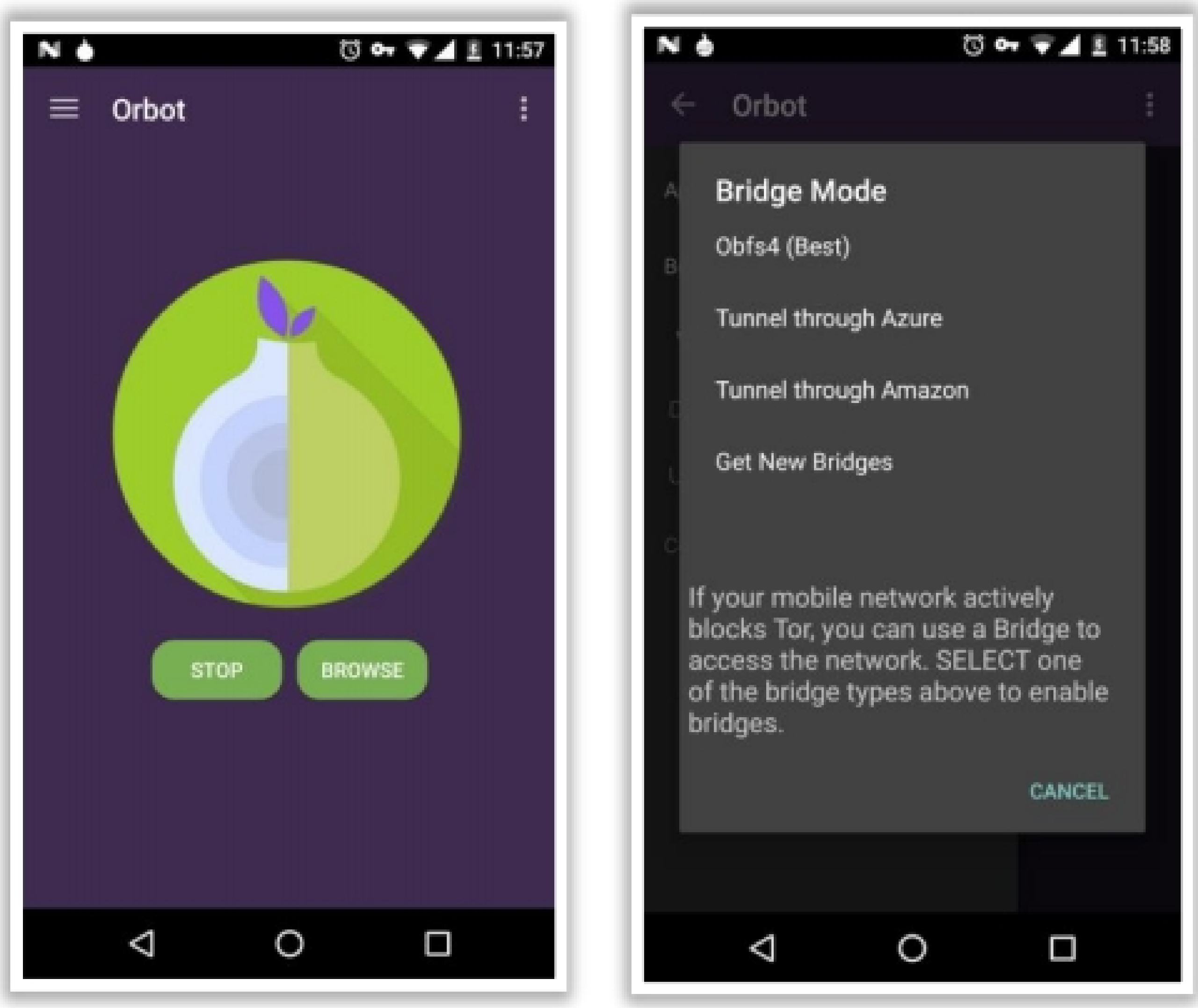
If your mobile network actively blocks Tor, you can use a Bridge to access the network. SELECT one of the bridge types above to enable bridges.

https://guardianproject.info

Hacking with Orbot Proxy

Source: *https://guardianproject.info*

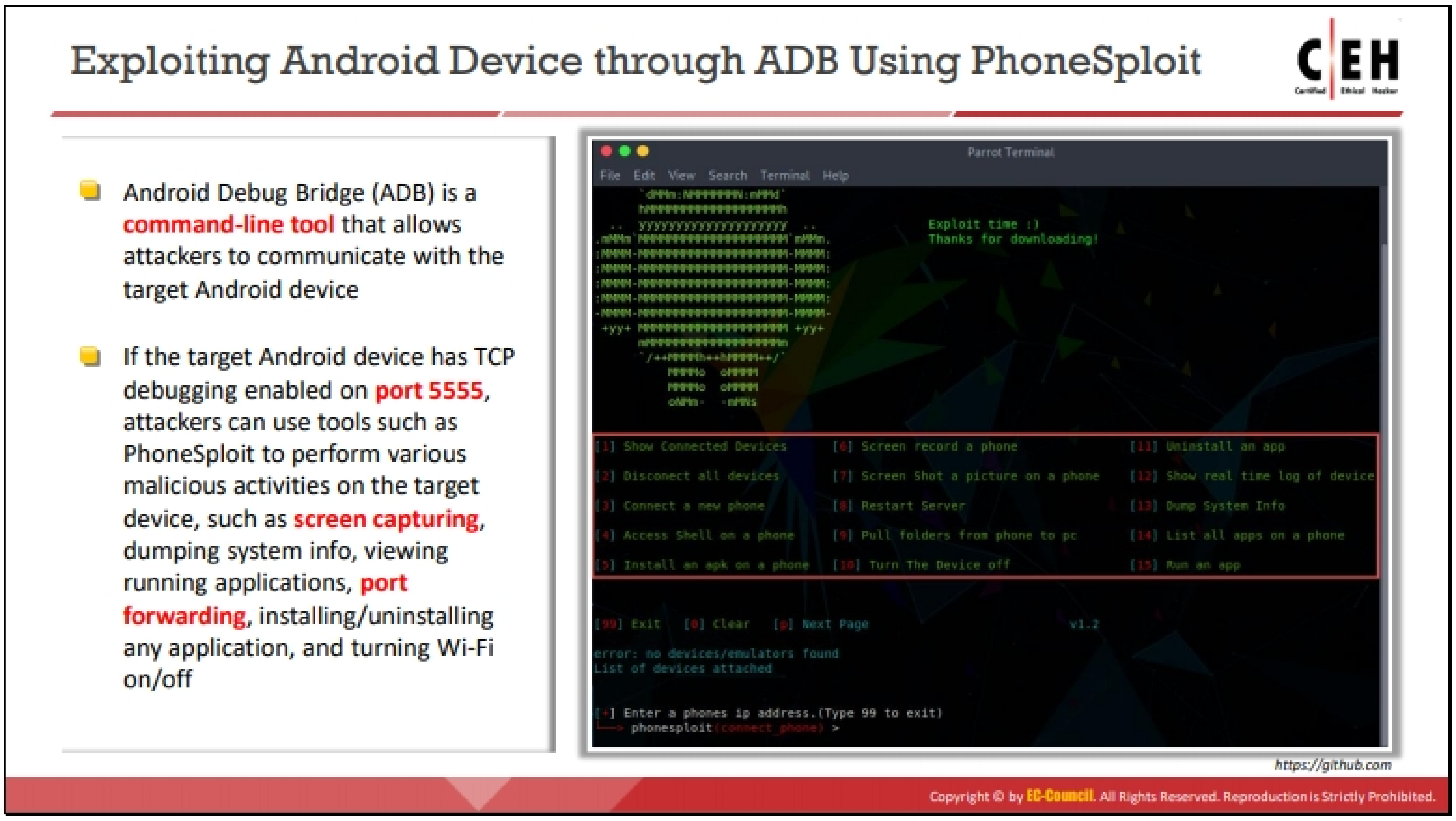
Orbot is a proxy app that empowers other apps to use the Internet more privately. It uses Tor to encrypt your Internet traffic and then hides it by bouncing it through a series of computers around the world. Attackers can use this application to hide their identity while performing attacks or surfing through target web applications.



Orbot
STOP
BROWSE
Orbot
Bridge Mode
Obfs4 (Best)
Tunnel through Azure
Tunnel through Amazon
Get New Bridges
CANCEL

If your mobile network actively blocks Tor, you can use a Bridge to access the network. SELECT one of the bridge types above to enable bridges.

Figure 17.26: Screenshot of Orbot Proxy



Exploiting Android Device through ADB Using PhoneSploit

- Android Debug Bridge (ADB) is a **command-line tool** that allows attackers to communicate with the target Android device
- If the target Android device has TCP debugging enabled on **port 5555**, attackers can use tools such as PhoneSploit to perform various malicious activities on the target device, such as **screen capturing**, dumping system info, viewing running applications, **port forwarding**, installing/uninstalling any application, and turning Wi-Fi on/off

```
                                          Exploit time :)
                                          Thanks for downloading!

[1] Show Connected Devices      [6] Screen record a phone                [11] Uninstall an app

[2] Disconect all devices       [7] Screen Shot a picture on a phone     [12] Show real time log of device

[3] Connect a new phone         [8] Restart Server                       [13] Dump System Info

[4] Access Shell on a phone     [9] Pull folders from phone to pc        [14] List all apps on a phone

[5] Install an apk on a phone   [10] Turn The Device off                 [15] Run an app


[99] Exit   [0] Clear   [p] Next Page                         v1.2

error: no devices/emulators found
List of devices attached


[+] Enter a phones ip address.(Type 99 to exit)
└──> phonesploit(connect_phone) >
```

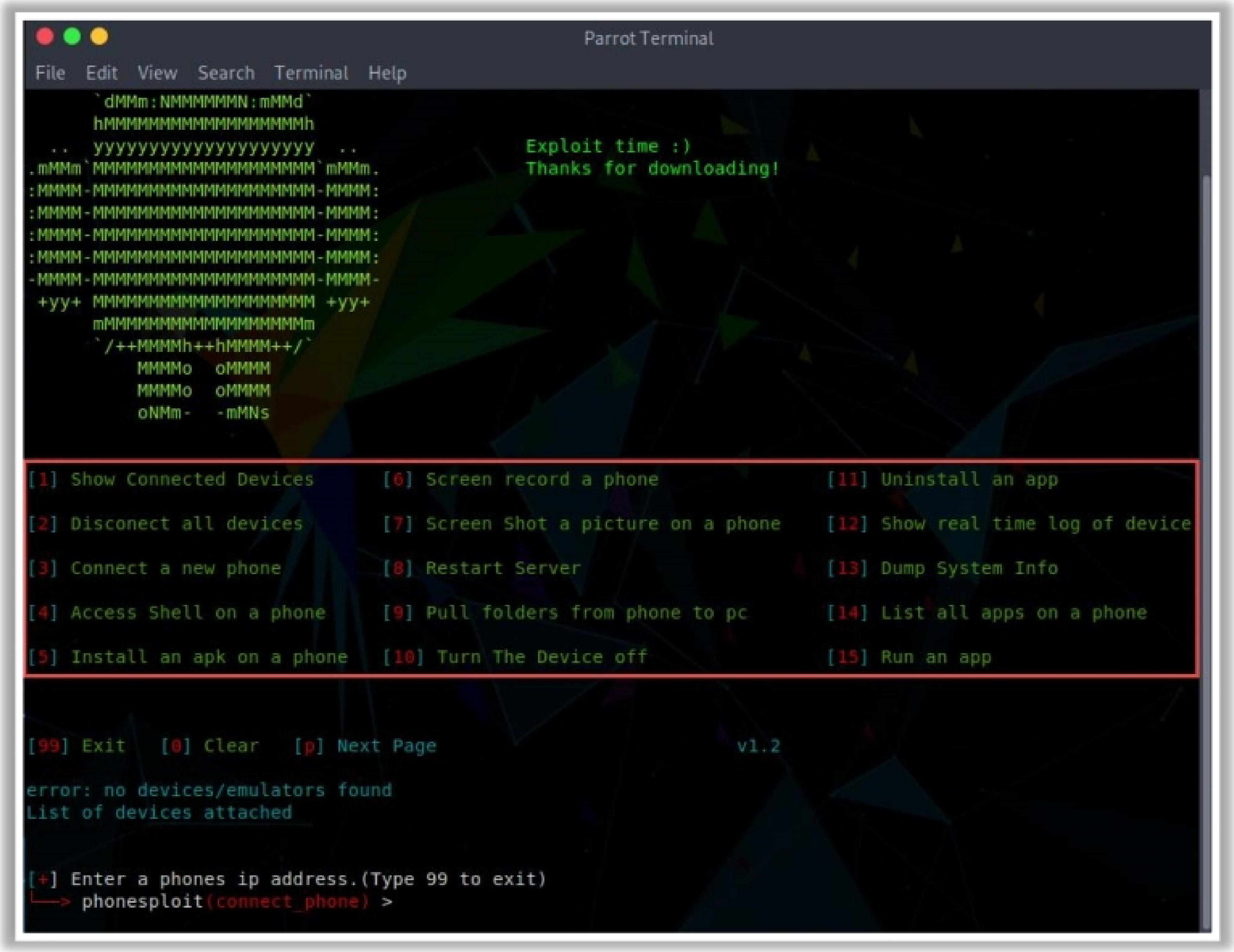
https://github.com

Exploiting Android Device through ADB Using PhoneSploit

Source: *https://github.com*

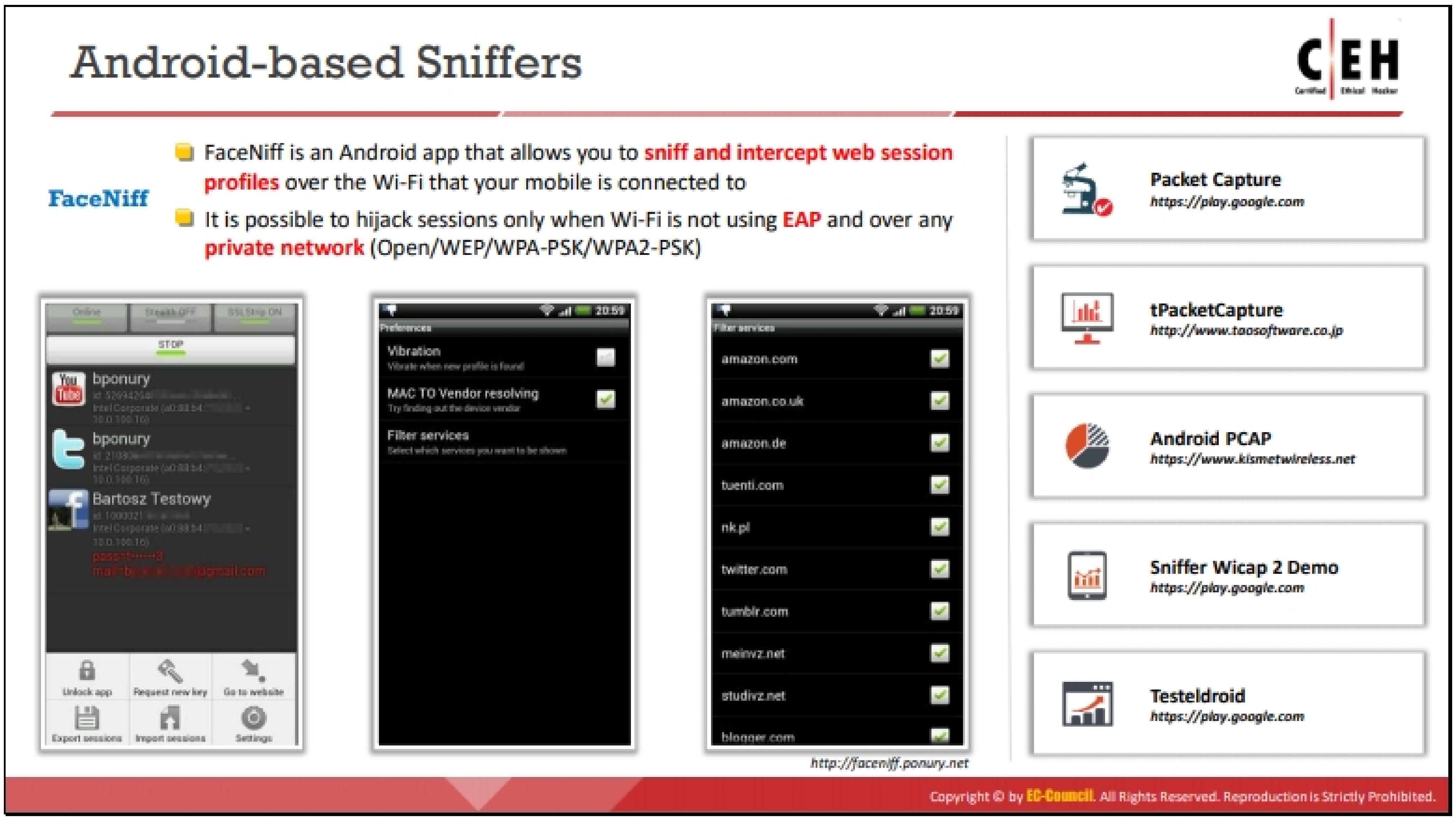
Android Debug Bridge (ADB) is a command-line tool that allows attackers to communicate with the target Android device. This tool provides various features to install and debug apps and access the Unix shell to execute various shell commands on a device. ADB service can be connected using a USB cable or ADB wireless. To use ADB wireless connectivity, you need to enable the daemon server using TCP port 5555 on the target device. This tool acts as a bridge between the attacker's PC and the target Android device. Furthermore, it provides a command window to run the commands directly on the Android device.

If the target Android device has TCP debugging enabled on port 5555, attackers can use tools such as PhoneSploit to perform various malicious activities on the target device, such as screen capture, dumping system info, viewing running applications, port forwarding, installing/uninstalling any application, and turning Wi-Fi On/Off.



```
      `dMMm:NMMMMMMN:mMMd`
      hMMMMMMMMMMMMMMMMMMh
  ..  yyyyyyyyyyyyyyyyyyyy  ..                Exploit time :)
.mMMm`MMMMMMMMMMMMMMMMMMMM`mMMm.              Thanks for downloading!
:MMMM-MMMMMMMMMMMMMMMMMMMM-MMMM:
:MMMM-MMMMMMMMMMMMMMMMMMMM-MMMM:
:MMMM-MMMMMMMMMMMMMMMMMMMM-MMMM:
:MMMM-MMMMMMMMMMMMMMMMMMMM-MMMM:
-MMMM-MMMMMMMMMMMMMMMMMMMM-MMMM-
 +yy+ MMMMMMMMMMMMMMMMMMMM +yy+
      mMMMMMMMMMMMMMMMMMMm
      `/++MMMMh++hMMMM++/`
          MMMMo  oMMMM
          MMMMo  oMMMM
          oNMm-  -mMNs

[1] Show Connected Devices     [6] Screen record a phone               [11] Uninstall an app

[2] Disconect all devices      [7] Screen Shot a picture on a phone    [12] Show real time log of device

[3] Connect a new phone        [8] Restart Server                      [13] Dump System Info

[4] Access Shell on a phone    [9] Pull folders from phone to pc       [14] List all apps on a phone

[5] Install an apk on a phone  [10] Turn The Device off                [15] Run an app


[99] Exit   [0] Clear   [p] Next Page                          v1.2

error: no devices/emulators found
List of devices attached


[+] Enter a phones ip address.(Type 99 to exit)
└──> phonesploit(connect_phone) >
```

Figure 17.27: Screenshot of PhoneSploit



Android-based Sniffers

FaceNiff

- FaceNiff is an Android app that allows you to **sniff and intercept web session profiles** over the Wi-Fi that your mobile is connected to
- It is possible to hijack sessions only when Wi-Fi is not using **EAP** and over any **private network** (Open/WEP/WPA-PSK/WPA2-PSK)

http://faceniff.ponury.net

Packet Capture
https://play.google.com

tPacketCapture
http://www.taosoftware.co.jp

Android PCAP
https://www.kismetwireless.net

Sniffer Wicap 2 Demo
https://play.google.com

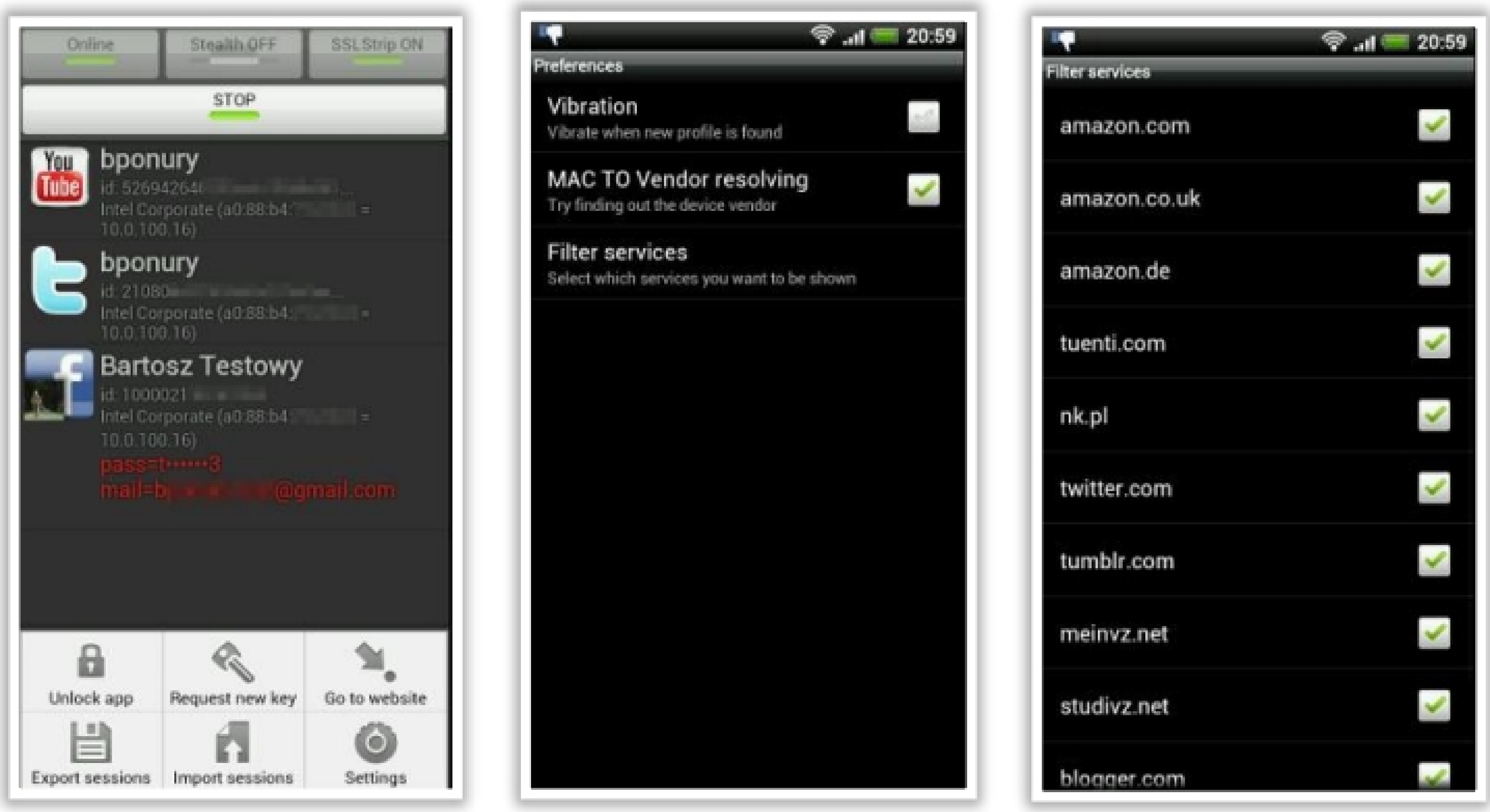
Testeldroid
https://play.google.com

Android-based Sniffers

- **FaceNiff**

  Source: *http://faceniff.ponury.net*

  FaceNiff is an Android app that allows you to sniff and intercept web session profiles over the Wi-Fi network to which your mobile device is connected. It is possible to hijack sessions only when Wi-Fi is not using extensible authentication protocol (EAP), but it should work over any private network (Open/WEP/WPA-PSK/WPA2-PSK).



bponury
id: 52694264[redacted]
Intel Corporate (a0:88:b4:[redacted]) = 10.0.100.16

bponury
id: 21080[redacted]
Intel Corporate (a0:88:b4:[redacted]) = 10.0.100.16

Bartosz Testowy
id: 1000021[redacted]
Intel Corporate (a0:88:b4:[redacted]) = 10.0.100.16
pass=t[redacted]3
mail=bj[redacted]@gmail.com

Preferences

Vibration ☐
Vibrate when new profile is found

MAC TO Vendor resolving ☑
Try finding out the device vendor

Filter services
Select which services you want to be shown

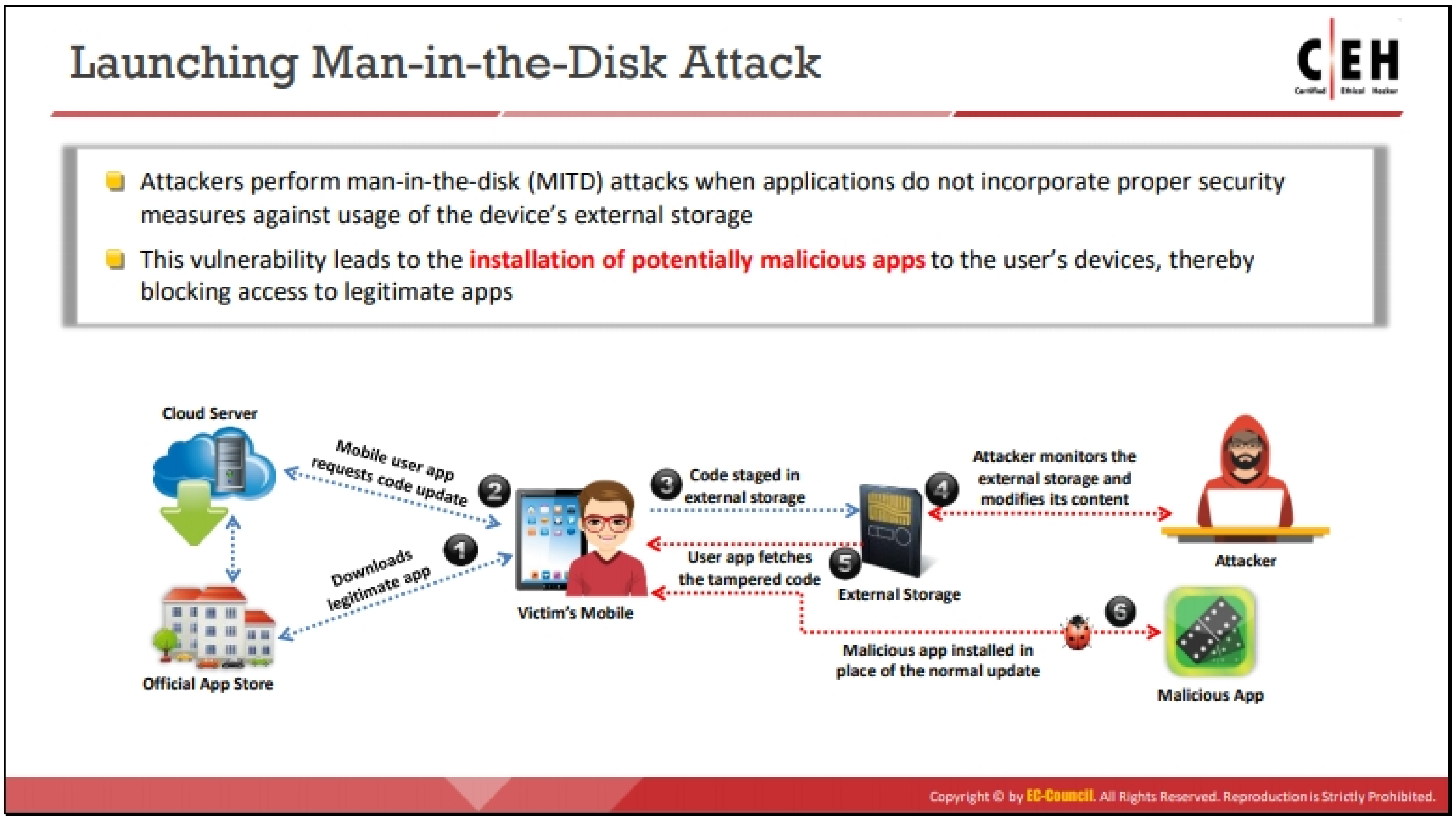
Filter services

amazon.com ☑
amazon.co.uk ☑
amazon.de ☑
tuenti.com ☑
nk.pl ☑
twitter.com ☑
tumblr.com ☑
meinvz.net ☑
studivz.net ☑
blogger.com ☑

Figure 17.28: Screenshots of FaceNiff

Some additional Android-based sniffers are as follows:

- Packet Capture (*https://play.google.com*)
- tPacketCapture (*http://www.taosoftware.co.jp*)
- Android PCAP (*https://www.kismetwireless.net*)
- Sniffer Wicap 2 Demo (*https://play.google.com*)
- Testeldroid (*https://play.google.com*)



Launching Man-in-the-Disk Attack

- Attackers perform man-in-the-disk (MITD) attacks when applications do not incorporate proper security measures against usage of the device's external storage
- This vulnerability leads to the **installation of potentially malicious apps** to the user's devices, thereby blocking access to legitimate apps

Cloud Server
Mobile user app requests code update
2
Downloads legitimate app
1
Official App Store
Victim's Mobile
3
Code staged in external storage
User app fetches the tampered code
5
External Storage
4
Attacker monitors the external storage and modifies its content
Attacker
6
Malicious app installed in place of the normal update
Malicious App

Launching Man-in-the-Disk Attack

Attackers perform a man-in-the-disk (MITD) attack when applications do not incorporate proper security measures against the usage of the device's external storage. This vulnerability leads to the installation of potentially malicious apps on the user's device, thereby blocking access to legitimate apps. MITD is a variation of MITM. The Android OS consists of two types of storage: internal and external. In general, the internal storage for Android apps is sandboxed, whereas the external storage is envisioned to allow file sharing between apps, making it vulnerable to MITD attacks.

When any legitimate app tries to run a regular update, an attacker monitors the data stored in the external storage and tries to replace, manipulate, or overwrite the application data by tampering with the source code of the update. After the attacker successfully injects malicious code into the legitimate app update, the user app fetches and runs the malicious code and installs a fraudulent app from the attacker.

Using this malicious app, the attacker can evade Android security and gain access to the sensitive information stored on the device, such as login credentials, personal information, contacts, and photos, and even hack mobile hardware such as microphones and cameras. This malicious app can further cause the application to blackout and then completely take control of the mobile device.

An MITD attack involves the following steps:

- Victim downloads and installs a legitimate app from the official app store
- Victim's mobile device receives an app update and requests code update from the cloud server

- Victim gives permission to the legitimate app to access external storage. Now, the downloaded code is stored on the external storage
- Attacker remotely monitors the external storage and modifies its content by injecting the malicious code
- Now, the legitimate app fetches and runs the tampered update code from the external storage
- The malicious code injected by the attacker automatically requests and installs a fraudulent app from the attacker.
- Using this malicious app, the attacker can steal the victim's sensitive information stored on the mobile device or completely take control of the mobile device.

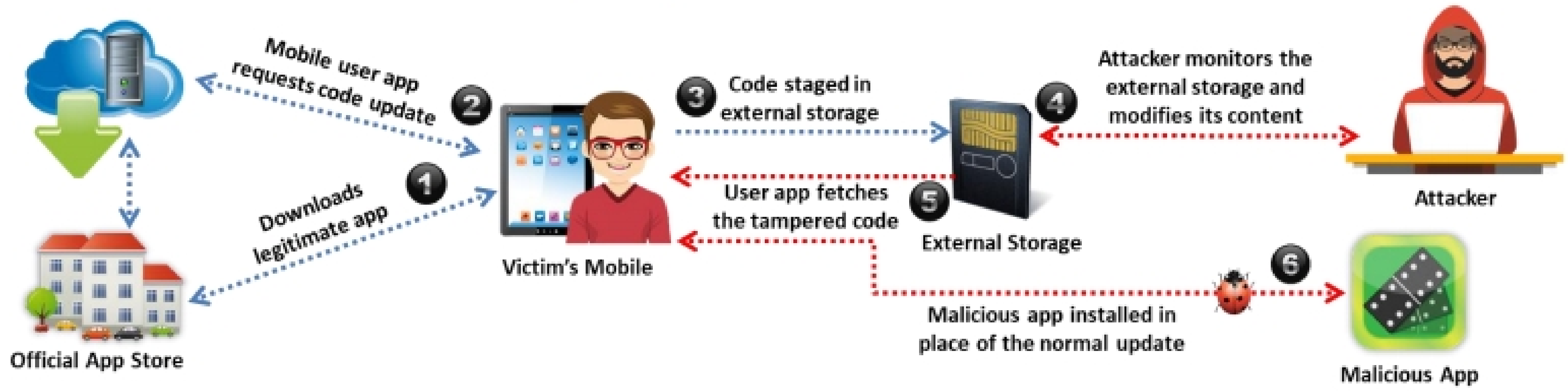
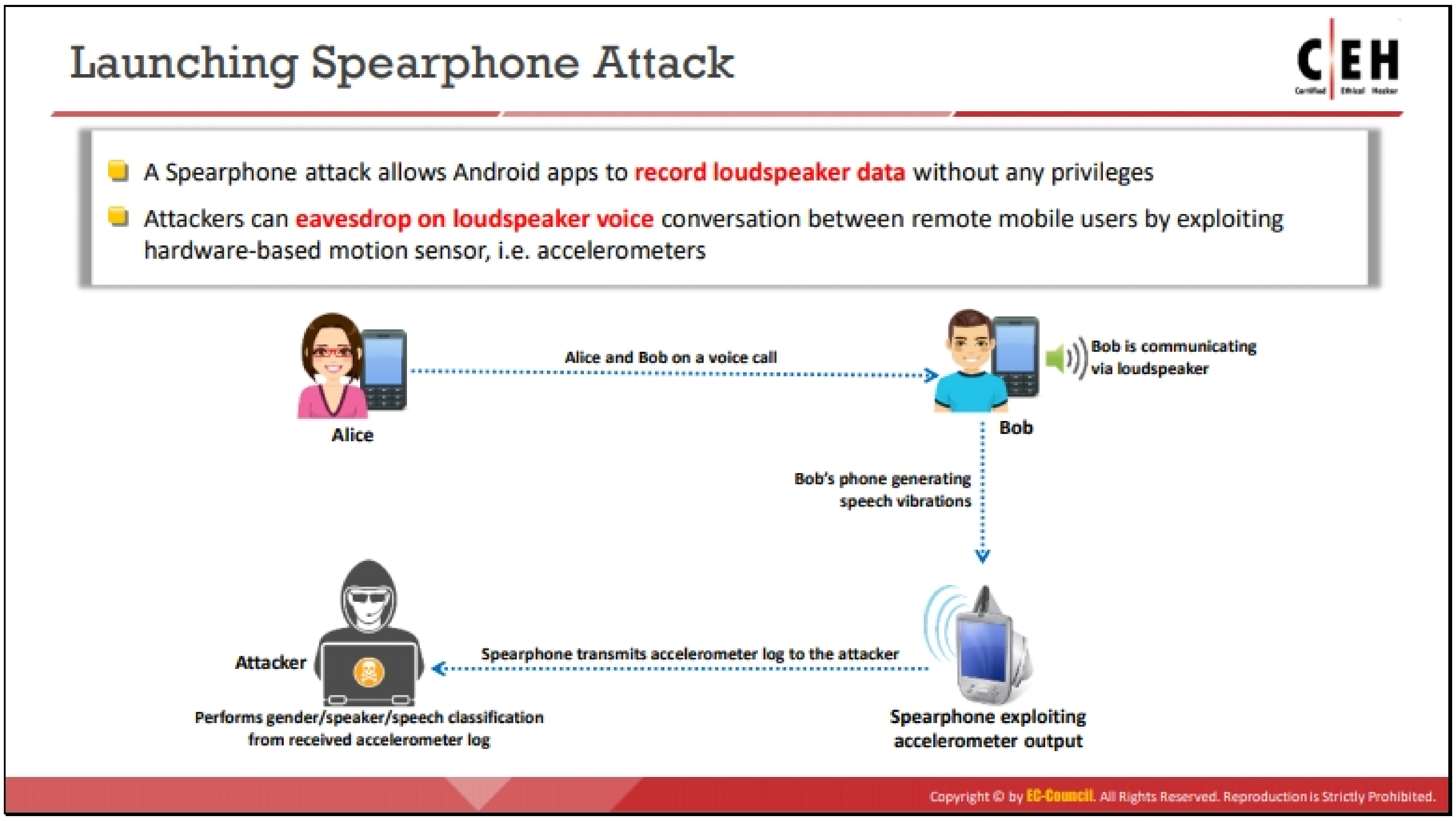

Mobile user app requests code update
②
Downloads legitimate app
①
Official App Store
Victim's Mobile
③ Code staged in external storage
User app fetches the tampered code ⑤
External Storage
④ Attacker monitors the external storage and modifies its content
Attacker
Malicious app installed in place of the normal update
⑥
Malicious App

Figure 17.29: Man-in-the-disk attack



Launching Spearphone Attack

- A Spearphone attack allows Android apps to **record loudspeaker data** without any privileges
- Attackers can **eavesdrop on loudspeaker voice** conversation between remote mobile users by exploiting hardware-based motion sensor, i.e. accelerometers

Alice and Bob on a voice call
Bob is communicating via loudspeaker
Alice
Bob
Bob's phone generating speech vibrations
Attacker
Spearphone transmits accelerometer log to the attacker
Performs gender/speaker/speech classification from received accelerometer log
Spearphone exploiting accelerometer output

Launching Spearphone Attack

A spearphone attack allows Android apps to record loudspeaker data without any privileges. Attackers can eavesdrop on loudspeaker voice conversations between remote mobile users by exploiting the hardware-based motion sensor, i.e., the accelerometer. The accelerometer is a firmware chip embedded in most smartphones, and it can be accessed by any app installed on the phone with no special permissions. The motion sensor allows apps to capture the physical movement of the device based on the changes in position and velocity. Speech reverberations can also be recorded through this built-in sensor, as the loudspeaker is placed on the same surface in the device.

Attackers can also monitor the loudspeaker's output data such as voice assistants, multimedia messages, and audio files, using a malicious app to breach speech privacy. In addition, attackers can capture data using malicious code running on the phone. Furthermore, they can perform speech or speaker identification and gender classification by implementing speech recognition and reconstruction. The diagram below shows how loudspeaker data are captured.

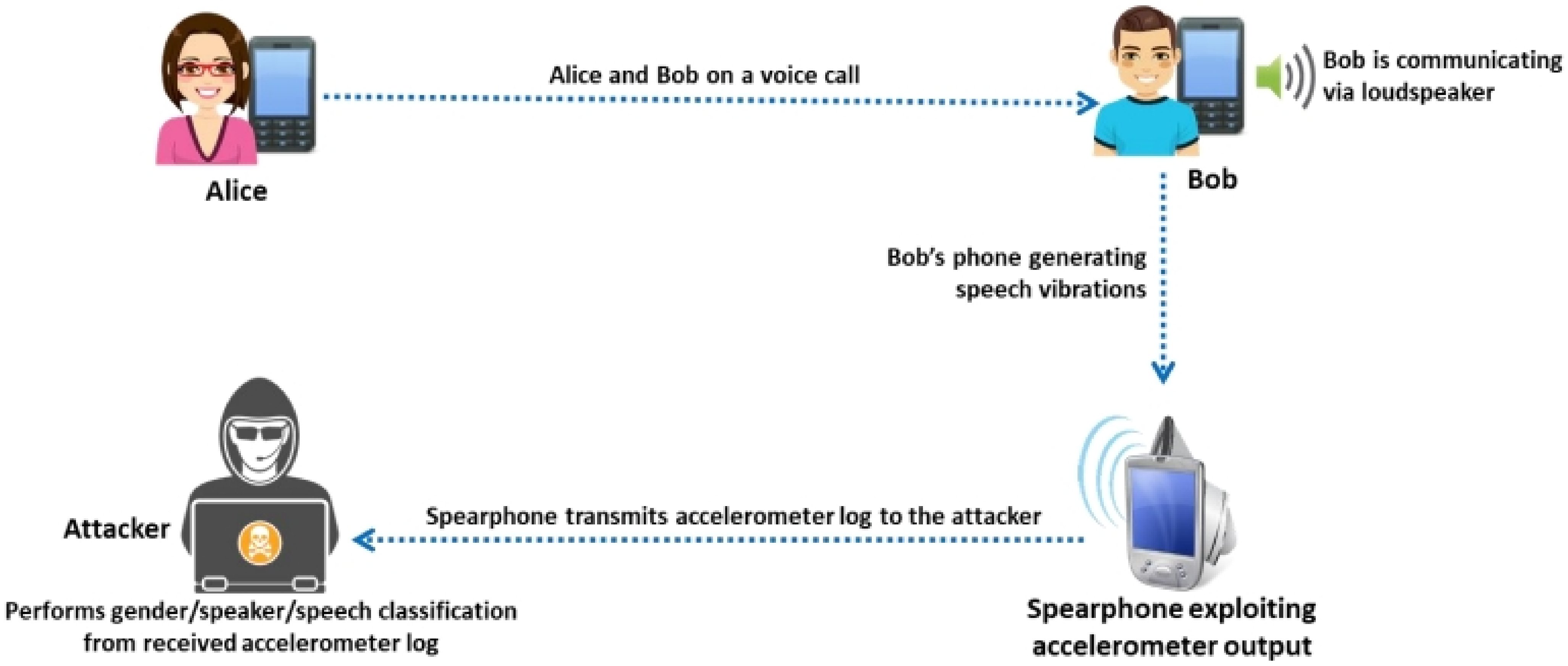
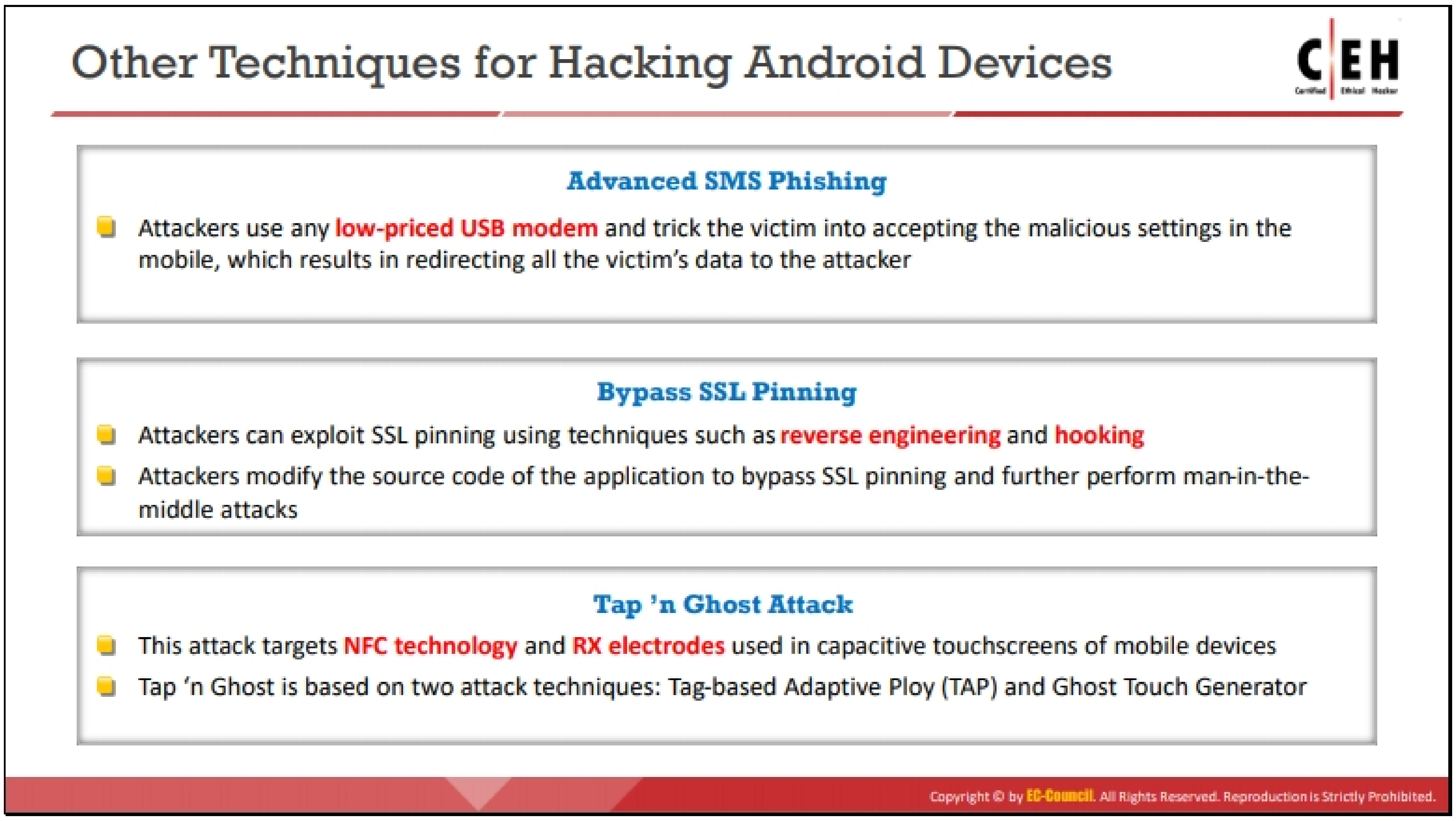

Alice and Bob on a voice call
Bob is communicating via loudspeaker
Alice
Bob
Bob's phone generating speech vibrations
Attacker
Spearphone transmits accelerometer log to the attacker
Performs gender/speaker/speech classification from received accelerometer log
Spearphone exploiting accelerometer output

Figure 17.30: Spearphone attack



Other Techniques for Hacking Android Devices

Advanced SMS Phishing

- Attackers use any **low-priced USB modem** and trick the victim into accepting the malicious settings in the mobile, which results in redirecting all the victim's data to the attacker

Bypass SSL Pinning

- Attackers can exploit SSL pinning using techniques such as **reverse engineering** and **hooking**
- Attackers modify the source code of the application to bypass SSL pinning and further perform man-in-the-middle attacks

Tap 'n Ghost Attack

- This attack targets **NFC technology** and **RX electrodes** used in capacitive touchscreens of mobile devices
- Tap 'n Ghost is based on two attack techniques: Tag-based Adaptive Ploy (TAP) and Ghost Touch Generator

Other Techniques for Hacking Android Devices

- **Advanced SMS Phishing**

  Advanced SMS phishing attack is a type of phishing scam that occurs due to security flaws in the latest Android-based smartphones mostly manufactured by Samsung, Huawei, LG, and Sony. The attacker can perform this attack using any low-priced USB modem and tricking the user into accepting the new settings, i.e., malicious settings, in the mobile device, which can redirect the user's data to the attacker.

  The attack vector mainly depends on a process called Over-the-Air (OTA) provisioning, which is mainly used by network operators. OTA is a mechanism that is used to send provisioning data and updates in a mobile device remotely. Due to its weak authentication methods, OTA is easily vulnerable to phishing attacks. The attacker exploits the mobile device by sending messages that seem to be genuine from the network operator. These messages contain malicious links that can redirect Internet traffic back to the attacker.

  Before performing the attack, the attacker requires the victim's International Mobile Subscriber Identity (IMSI) number, which is a unique string identifier for every mobile device. Using this IMSI, the attacker's malicious message can be easily authenticated and processed in the mobile device. If the IMSI number is not available, the attacker should send two messages. The first malicious message contains a PIN that appears to be from the victim's network operator; the second message encompasses the malicious message authenticated with the PIN in the first message. Using these messages, the mobile device can be exploited when the victim enters the PIN. These types of messages containing malevolent links can modify the message servers, mail servers, directory servers, and proxy addresses of Android-based smartphones. SMiShing attacks can be mitigated using applications such as SandBlast Mobile.

- **Bypass SSL Pinning**

  SSL pinning allows applications to perform operations only after validating trusted certificates and public keys. Although the communication between the application and the server is defined by SSL pinning, which can prevent MITM attacks, an attacker can still bypass SSL pinning using various techniques by exploiting the misconfigurations in SSL implementation.

  These techniques include reverse engineering and hooking as well as various automated tools such as Apktool, Frida, keytool, and Jarsigner.

  - **Using Reverse Engineering**

    Attackers use tools such as Apktool to decompile and recompile the application to its original form after some modifications.

    Steps to bypass SSL pinning using reverse engineering:

    - Download Apktool (offers command-line interface)
    - Use the following command to decompile the Android application:

      ```
      apktool d <application_name.apk>
      ```

    - After decompiling the application, you can gain access to the APK source code along with different directories such as smali, build, smali_classes, assets, lib, unknown, original, and res.
    - smali (assembler used by Dalvik VM, an Android JVM implementation) code is an integral part of decompiled code that includes pre-installed Kotlin and Java code. You need to understand this application code to modify smali.
    - Now, you can try to discover SSL pinning that includes various functions such as checkclienttrust and checkclientserver, which provide information about X.509 digital certificates. These X.509 certificates contain the bytecode of the public key of the user. After gathering this information, you can alter the output of the function to bypass SSL pinning.

  - **Hooking**

    Using the hooking technique, an attacker can tamper with the runtime behavior of an application. Attacker uses tools such as Frida to alter the runtime code. Frida allows the attacker to inject malicious code into the application and manipulate the original code and working of the application.

- **Tap ’n Ghost Attack**

  Tap ’n Ghost is a novel attack technique that exploits NFC-enabled Android devices. This attack targets NFC technology and RX electrodes used in the capacitive touchscreens of mobile devices. If the attacker is able to establish a remote connection with the target mobile device, he/she can take full control of the device. Attackers use Bluetooth or Wi-Fi access points to establish a remote connection.

Tap ‘n Ghost is based on two attack techniques, namely Tag-based Adaptive Ploy (TAP) and Ghost Touch Generator. Attackers use these techniques to generate malicious events on the victim’s smartphone and take control of the smartphone remotely. Such attacks can also be launched on voting machines and ATMs.

- **Tag-based Adaptive Ploy (TAP)**

  TAP uses the NFC feature, which can trigger the Android device to visit a specific URL without the victim’s consent using the NFC tag emulator. This attack works with a web server that uses the device fingerprinting technique.

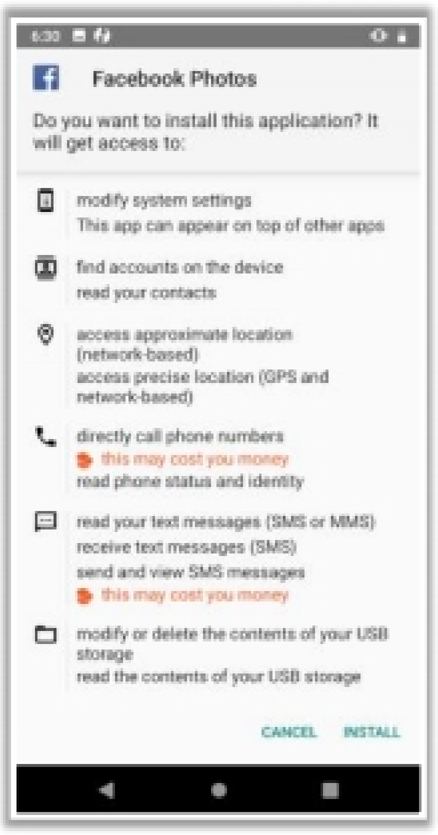
- **Ghost Touch Generator**

  Ghost Touch Generator works by forcing the victim to touch the cancel button, which does the work of the permit button. Thus, the attacker can trick the victim into granting remote access to the smartphone without the victim’s knowledge.

Android Trojans

Gustuff Banking Trojan

- Gustuff is a type of **banking Trojan** that uses malicious SMSs to compromise the security of target mobile devices
- Gustuff can **dynamically load webviews**, thereby targeting specific domains based on the received commands
- The potential targets of Gustuff include **Android-based banking apps**, online stores, messengers, and other crypto wallets



Facebook Photos

Do you want to install this application? It will get access to:

modify system settings
This app can appear on top of other apps

find accounts on the device
read your contacts

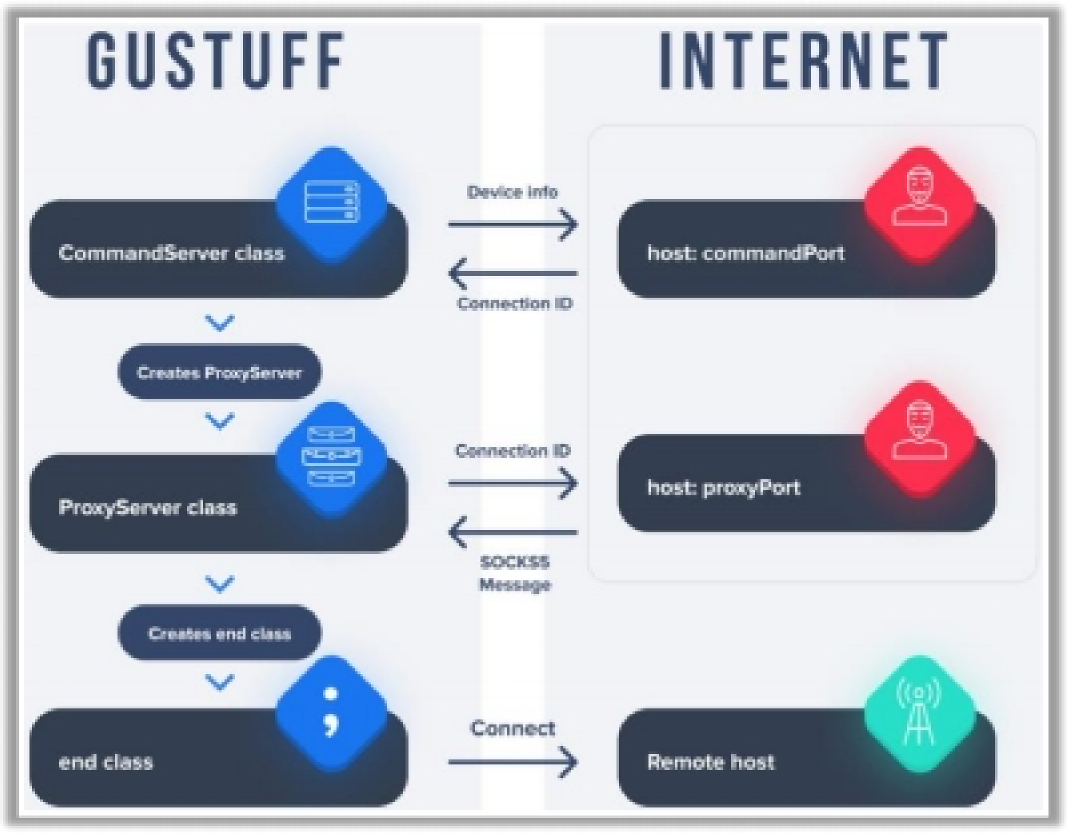
access approximate location (network-based)
access precise location (GPS and network-based)

directly call phone numbers
this may cost you money
read phone status and identity

read your text messages (SMS or MMS)
receive text messages (SMS)
send and view SMS messages
this may cost you money

modify or delete the contents of your USB storage
read the contents of your USB storage

6:30
CANCEL
INSTALL



GUSTUFF
INTERNET
CommandServer class
Device info
Connection ID
host: commandPort
Creates ProxyServer
ProxyServer class
Connection ID
SOCKS5 Message
host: proxyPort
Creates end class
end class
Connect
Remote host

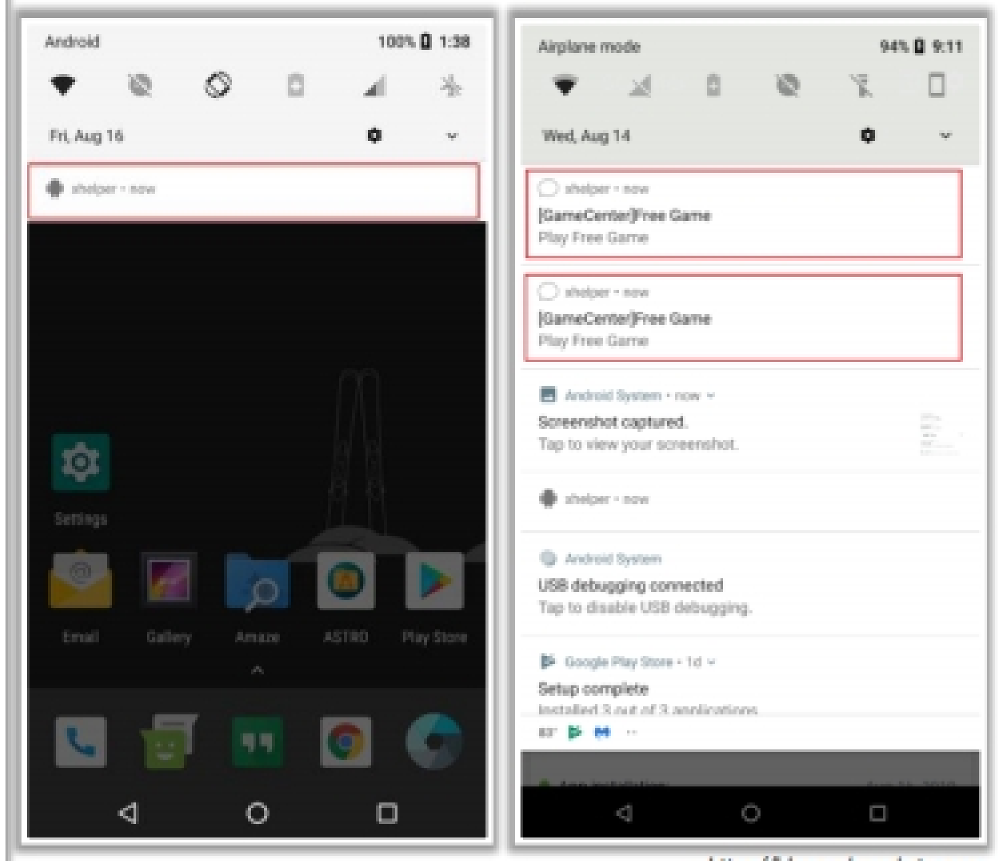
https://brica.de, https://www.group-ib.com

Android Trojans (Cont'd)

xHelper

On installation, the behavior of semi-stealth xHelper is as follows:

- Creates an icon in notifications titled "**xhelper**"
- Does not create an app icon or shortcut icon
- After a couple of minutes, more icons are added to the notifications: [GameCenter] Free Game
  - Press on either of these notifications; you will be redirected to a website that allows you to play games directly from your browser
  - These websites seem harmless, but malware authors are certainly collecting pay-for-click profits on each redirect



xhelper • now

xhelper • now
[GameCenter]Free Game
Play Free Game

xhelper • now
[GameCenter]Free Game
Play Free Game

Android System • now
Screenshot captured.
Tap to view your screenshot.

xhelper • now

Android System
USB debugging connected
Tap to disable USB debugging.

Google Play Store • 1d
Setup complete
Installed 3 out of 3 applications

Android
100% 1:38
Fri, Aug 16
Settings
Email
Gallery
Amaze
ASTRO
Play Store
Airplane mode
94% 9:11
Wed, Aug 14

https://blog.malwarebytes.com

- **Cerberus**
- **Boogr.gsh**
- **Asacub**
- **Gplayed**
- **HeroRat**

Android Trojans

- **Gustuff Banking Trojan**

  Source: *https://brica.de, https://www.group-ib.com*

  Gustuff is a type of banking Trojan that uses malicious SMS to compromise the security of the target mobile device. Gustuff can dynamically load webviews, targeting specific domains based on the received commands. During this process, it can also fetch the required injection from a remote server. It blocks a number of anti-virus and anti-malware tools to prevent detection. This Trojan has also been noted to ask the user to update his/her credit card information, which it then steals. It allows malicious actors to perform activities on the UI of the infected device. The potential targets of Gustuff include Android-based banking apps, online stores, messengers, and other crypto wallets.

Gustuff is installed on Android mobiles through SMS embedded with a malicious Android Package File (APK). When this package is installed, the target device is infected with Gustuff. Furthermore, based on the server's command, the infection spreads to other devices through the contact list or server database. Gustuff has a unique feature, namely Automatic Transfer Systems (ATS) that autofill the fields in legitimate banking apps and other apps to steal the user's credentials.

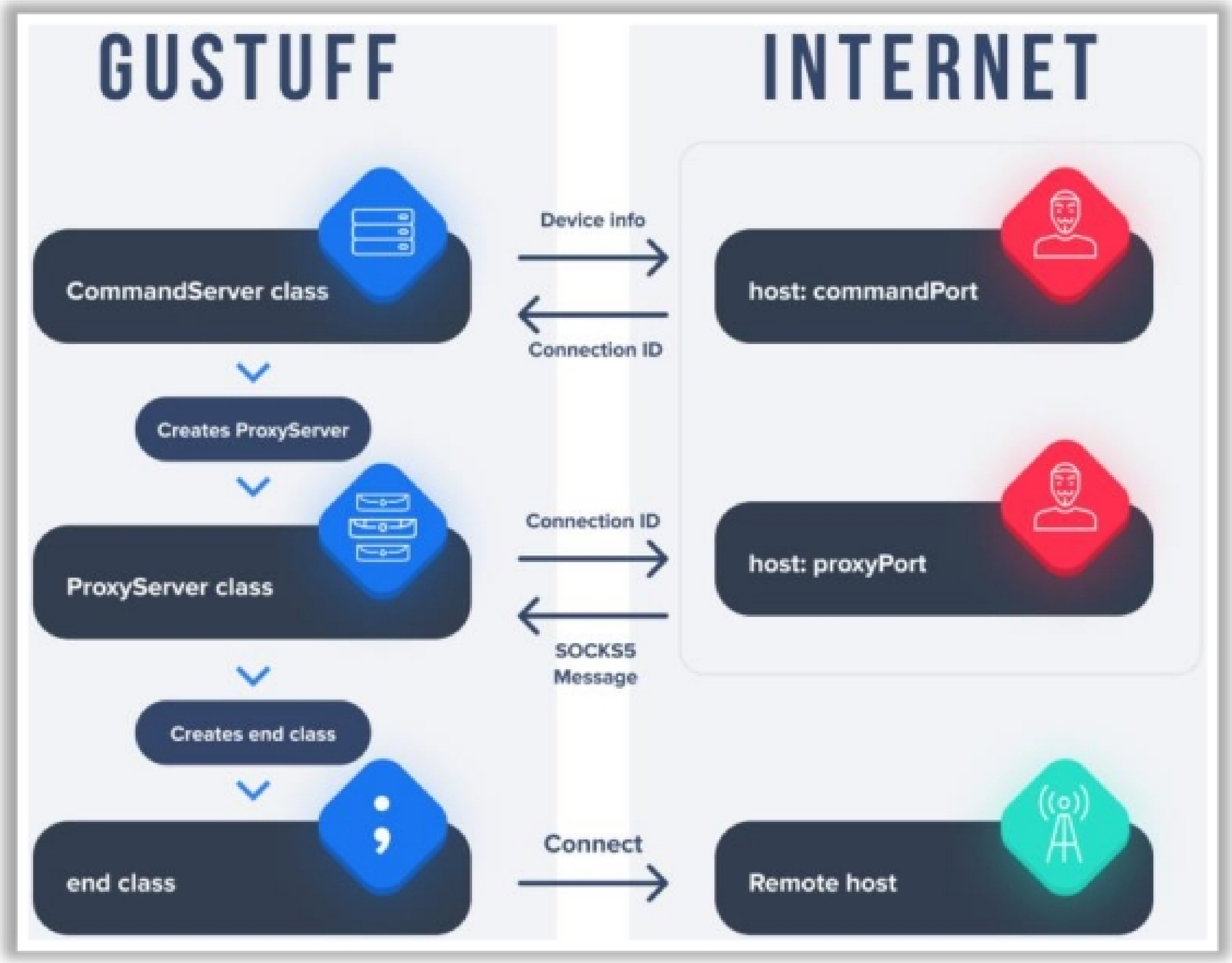
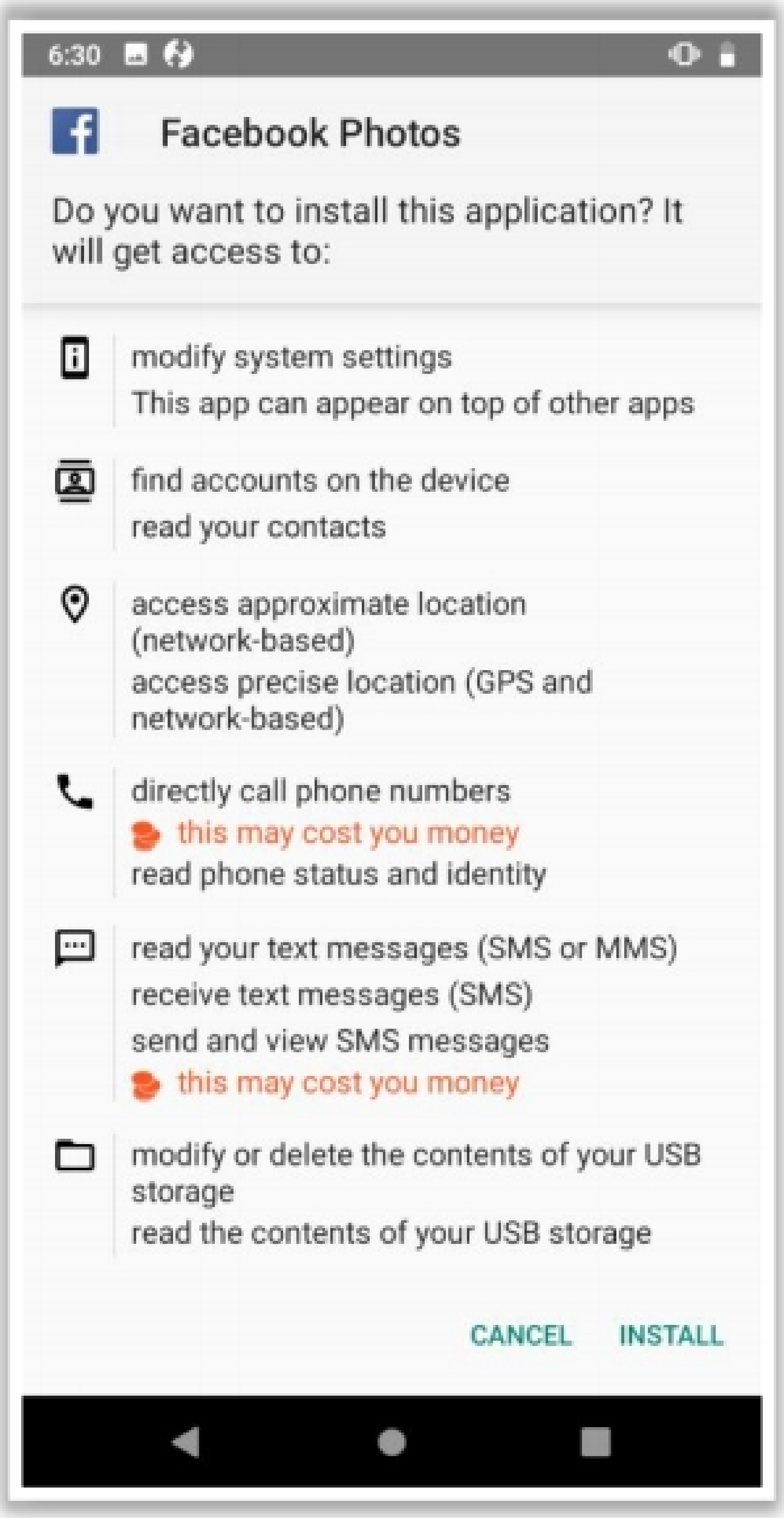

GUSTUFF
INTERNET
CommandServer class
Device info
Connection ID
host: commandPort
Creates ProxyServer
ProxyServer class
Connection ID
SOCKS5 Message
host: proxyPort
Creates end class
end class
Connect
Remote host

Figure 17.31: Working of Gustuff Trojan



Do you want to install this application? It will get access to:

modify system settings
This app can appear on top of other apps

find accounts on the device
read your contacts

access approximate location (network-based)
access precise location (GPS and network-based)

directly call phone numbers
this may cost you money
read phone status and identity

read your text messages (SMS or MMS)
receive text messages (SMS)
send and view SMS messages
this may cost you money

modify or delete the contents of your USB storage
read the contents of your USB storage

Figure 17.32: Screenshot of Gustuff Trojan

- **xHelper**

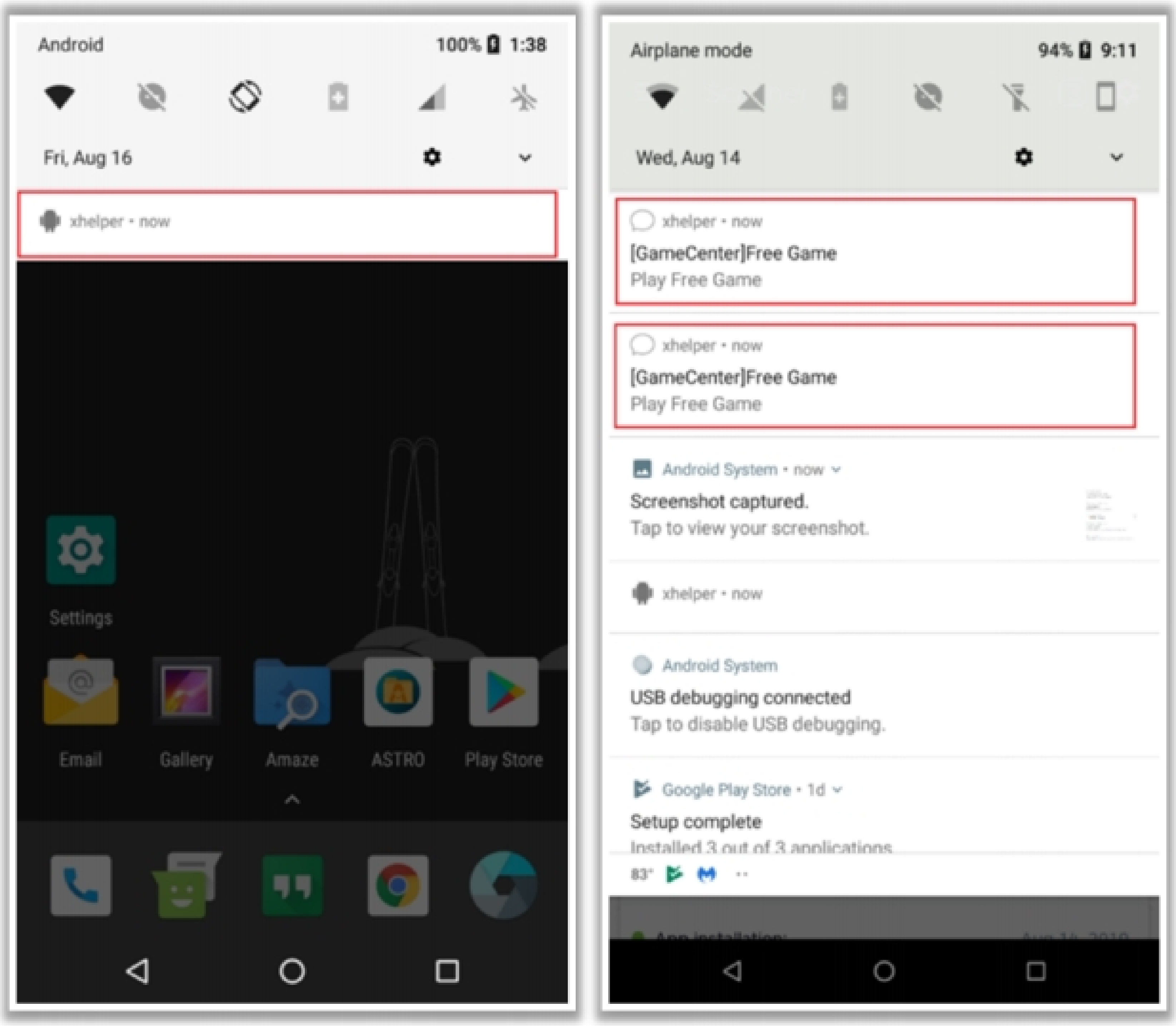
  Source: *https://blog.malwarebytes.com*

  Android/Trojan.Dropper.xHelper is a variant of Android/Trojan.Dropper. The first noticeable characteristic of xHelper is the use of stolen package names. For instance, xHelper uses package names starting with “com.muf.” This package name is associated with a number of puzzle games found on Google Play, including a puzzle called New2048HD with the package name com.mufc.fireuvw. This simple game had only a few more than 10 installs at the time of drafting this document.

  xHelper comes in two variants: full-stealth and semi-stealth. Upon installation, semi-stealth xHelper behaves as follows:

  1. Creates an icon in notifications titled “xhelper”
  2. Does not create an app icon or a shortcut icon
  3. After a couple of minutes, it starts adding more icons to notifications: [GameCenter] Free Game

- Press on either of these notifications, and it directs you to a website that allows you to play games directly via the browser.
- These websites seem harmless, but the malware authors are surely collecting pay-for-click profit on each redirect.


Android
100% 1:38
Fri, Aug 16
xhelper • now
Settings
Email
Gallery
Amaze
ASTRO
Play Store
Airplane mode
94% 9:11
Wed, Aug 14
xhelper • now
xhelper • now
Android System • now
xhelper • now
Android System
Google Play Store • 1d
83°

[GameCenter]Free Game
Play Free Game

[GameCenter]Free Game
Play Free Game

Screenshot captured.
Tap to view your screenshot.

USB debugging connected
Tap to disable USB debugging.

Setup complete
Installed 3 out of 3 applications

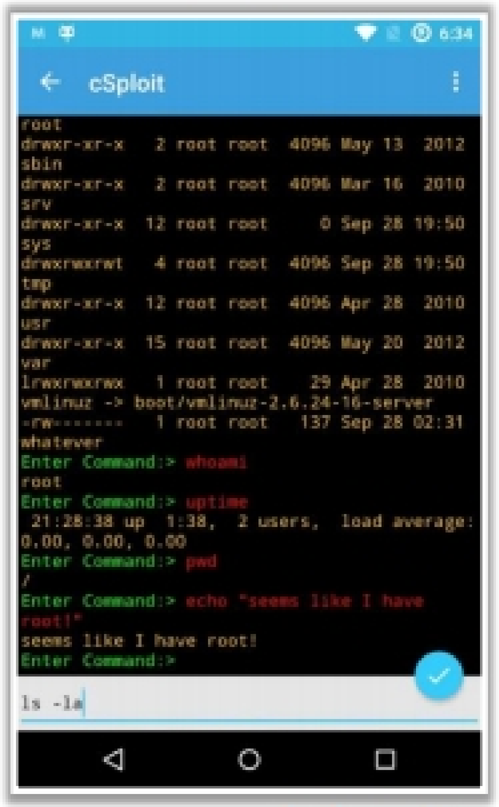
Figure 17.33: Screenshot of xHelper Trojan

Some additional Android Trojans are as follows:

- Cerberus
- Boogr.gsh
- Asacub
- Gplayed
- HeroRa

Android Hacking Tools

cSploit

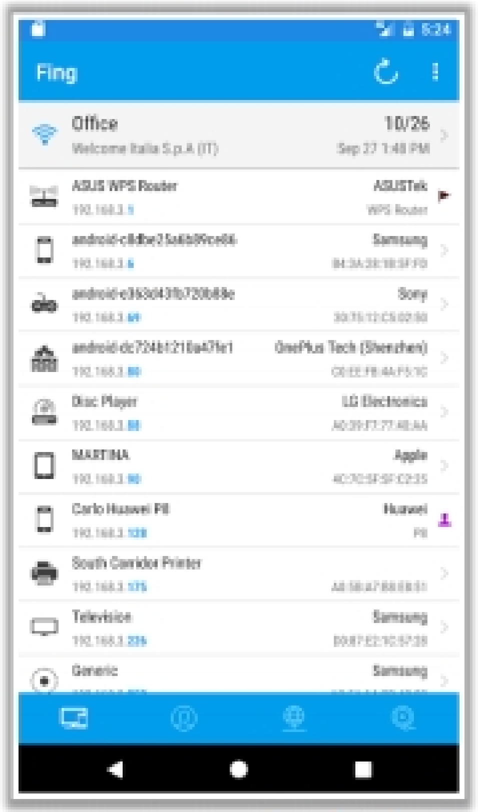


```
root
drwxr-xr-x   2 root root  4096 May 13  2012
sbin
drwxr-xr-x   2 root root  4096 Mar 16  2010
srv
drwxr-xr-x  12 root root     0 Sep 28 19:50
sys
drwxrwxrwt   4 root root  4096 Sep 28 19:50
tmp
drwxr-xr-x  12 root root  4096 Apr 28  2010
usr
drwxr-xr-x  15 root root  4096 May 20  2012
var
lrwxrwxrwx   1 root root    29 Apr 28  2010
vmlinuz -> boot/vmlinuz-2.6.24-16-server
-rw-------   1 root root   137 Sep 28 02:31
whatever
Enter Command:> whoami
root
Enter Command:> uptime
 21:28:38 up  1:38,  2 users,  load average:
0.00, 0.00, 0.00
Enter Command:> pwd
/
Enter Command:> echo "seems like I have
root!"
seems like I have root!
Enter Command:>
```

```
ls -la
```

http://www.csploit.org

Fing - Network Tools


Fing
Office
Welcome Italia S.p.A (IT)
10/26
Sep 27 1:48 PM
ASUS WPS Router
ASUSTek
WPS Router
Samsung
Sony
OnePlus Tech (Shenzhen)
Disc Player
LG Electronics
MARTINA
Apple
Carlo Huawei P8
Huawei
P8
South Corridor Printer
Television
Samsung
Generic
Samsung

https://www.fing.com

Androrat
https://github.com

Arpspoof
https://github.com

Network Discovery
https://github.com

NEXSPY
https://nexspy.com

IntentFuzzer
https://github.com

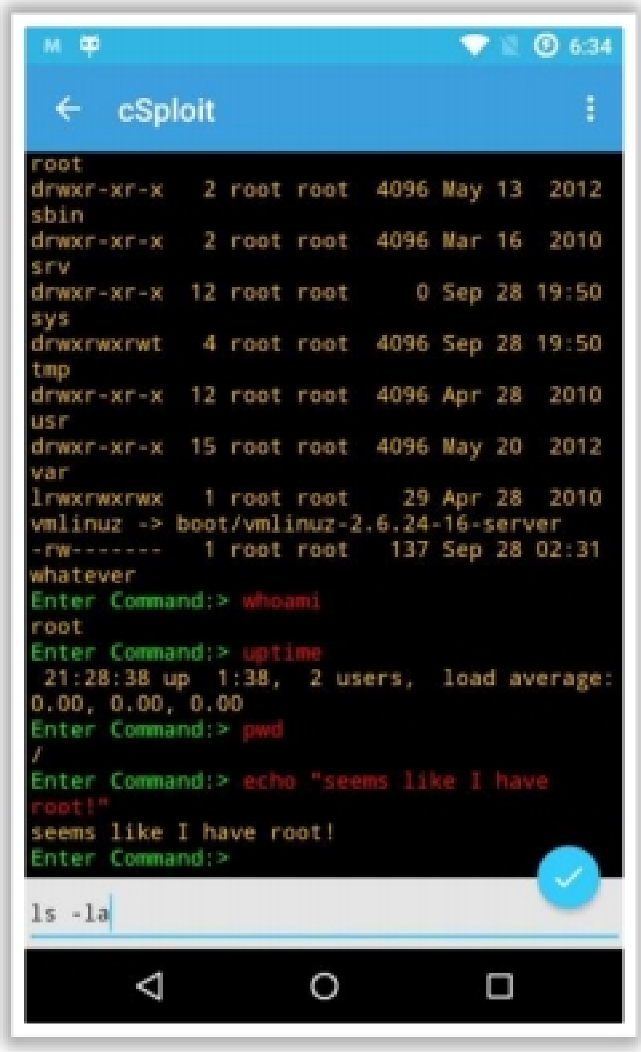
Android Hacking Tools

Attackers use various Android hacking tools to identify vulnerabilities and exploit target mobile devices to obtain critical user information such as credentials, personal information, and contact lists.

- **cSploit**

  Source: *http://www.csploit.org*

  cSploit is an Android network analysis and penetration suite that is used to map the local network, fingerprint hosts' operating systems and open ports, perform integrated traceroute, forge TCP/UDP packets, and perform MITM attacks such as password sniffing, JavaScript injection, capturing real-time network traffic, DNS spoofing, and session hijacking.



```
root
drwxr-xr-x   2 root root  4096 May 13  2012
sbin
drwxr-xr-x   2 root root  4096 Mar 16  2010
srv
drwxr-xr-x  12 root root     0 Sep 28 19:50
sys
drwxrwxrwt   4 root root  4096 Sep 28 19:50
tmp
drwxr-xr-x  12 root root  4096 Apr 28  2010
usr
drwxr-xr-x  15 root root  4096 May 20  2012
var
lrwxrwxrwx   1 root root    29 Apr 28  2010
vmlinuz -> boot/vmlinuz-2.6.24-16-server
-rw-------   1 root root   137 Sep 28 02:31
whatever
Enter Command:> whoami
root
Enter Command:> uptime
 21:28:38 up  1:38,  2 users,  load average:
0.00, 0.00, 0.00
Enter Command:> pwd
/
Enter Command:> echo "seems like I have
root!"
seems like I have root!
Enter Command:>
```

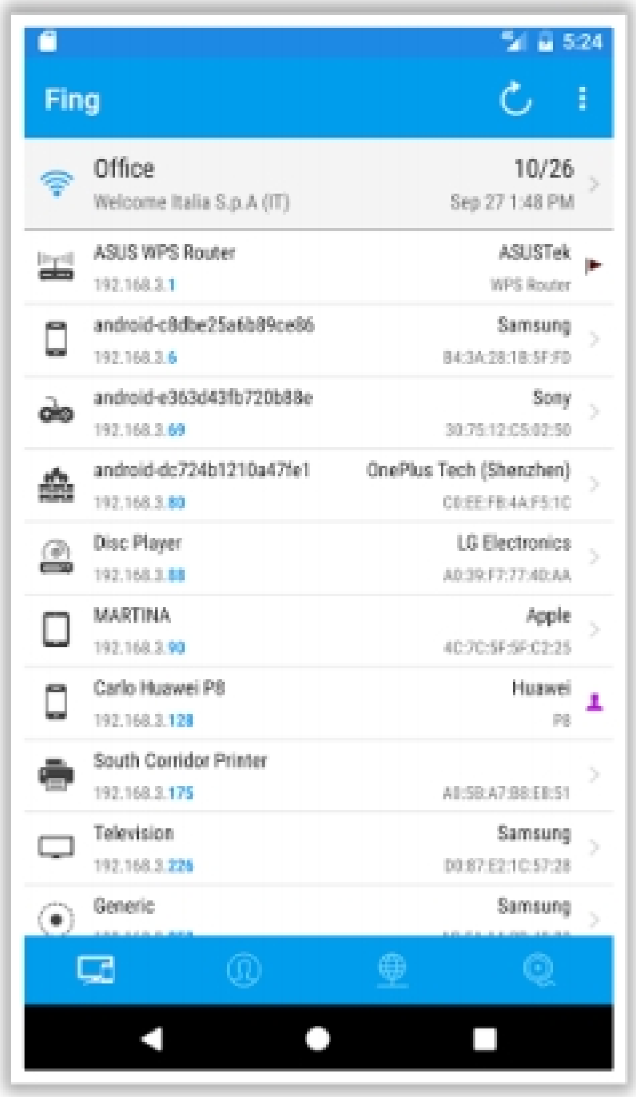
```
ls -la
```

Figure 17.34: Screenshot of cSploit

- ## Fing - Network Tools

  Source: *https://www.fing.com*

  Fing is a set of network tools that are used to identify all the devices connected to any network, obtain the IP address, MAC address, device name, model, and vendor of any connected device, retrieve advanced information such as NetBIOS, UPnP, Bonjour names, properties, and device types.



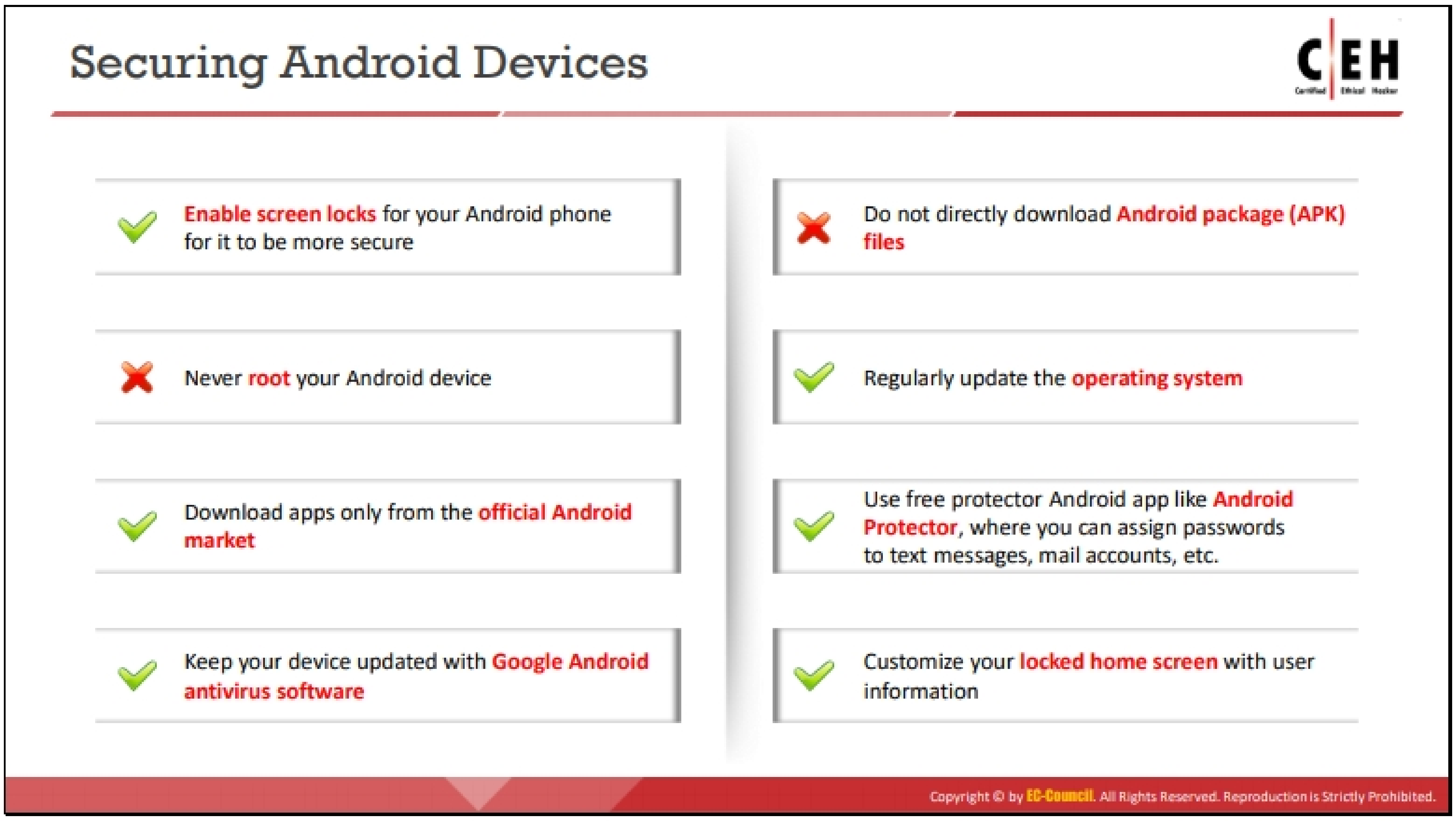
Office 10/26
Welcome Italia S.p.A (IT) Sep 27 1:48 PM

ASUS WPS Router	192.168.3.1	ASUSTek	WPS Router
android-c8dbe25a6b89ce86	192.168.3.6	Samsung	B4:3A:28:18:5F:FD
android-e363d43fb720b88e	192.168.3.69	Sony	30:75:12:C5:02:50
android-dc724b1210a47fe1	192.168.3.80	OnePlus Tech (Shenzhen)	C0:EE:FB:4A:F5:1C
Disc Player	192.168.3.88	LG Electronics	A0:39:F7:77:40:AA
MARTINA	192.168.3.90	Apple	4C:7C:5F:5F:C2:25
Carlo Huawei P8	192.168.3.128	Huawei	P8
South Corridor Printer	192.168.3.175		A0:5B:A7:B8:E8:51
Television	192.168.3.226	Samsung	D0:87:E2:1C:57:28
Generic		Samsung	

Figure 17.35: Screenshot of Fing - Network Tools

Some additional Android hacking tools are as follows:

- Androrat (*https://github.com*)
- Arpspoof (*https://github.com*)
- Network Discovery (*https://github.com*)
- NEXSPY (*https://nexspy.com*)
- IntentFuzzer (*https://github.com*)



Securing Android Devices

- ✓ **Enable screen locks** for your Android phone for it to be more secure
- ✗ Never **root** your Android device
- ✓ Download apps only from the **official Android market**
- ✓ Keep your device updated with **Google Android antivirus software**
- ✗ Do not directly download **Android package (APK) files**
- ✓ Regularly update the **operating system**
- ✓ Use free protector Android app like **Android Protector**, where you can assign passwords to text messages, mail accounts, etc.
- ✓ Customize your **locked home screen** with user information

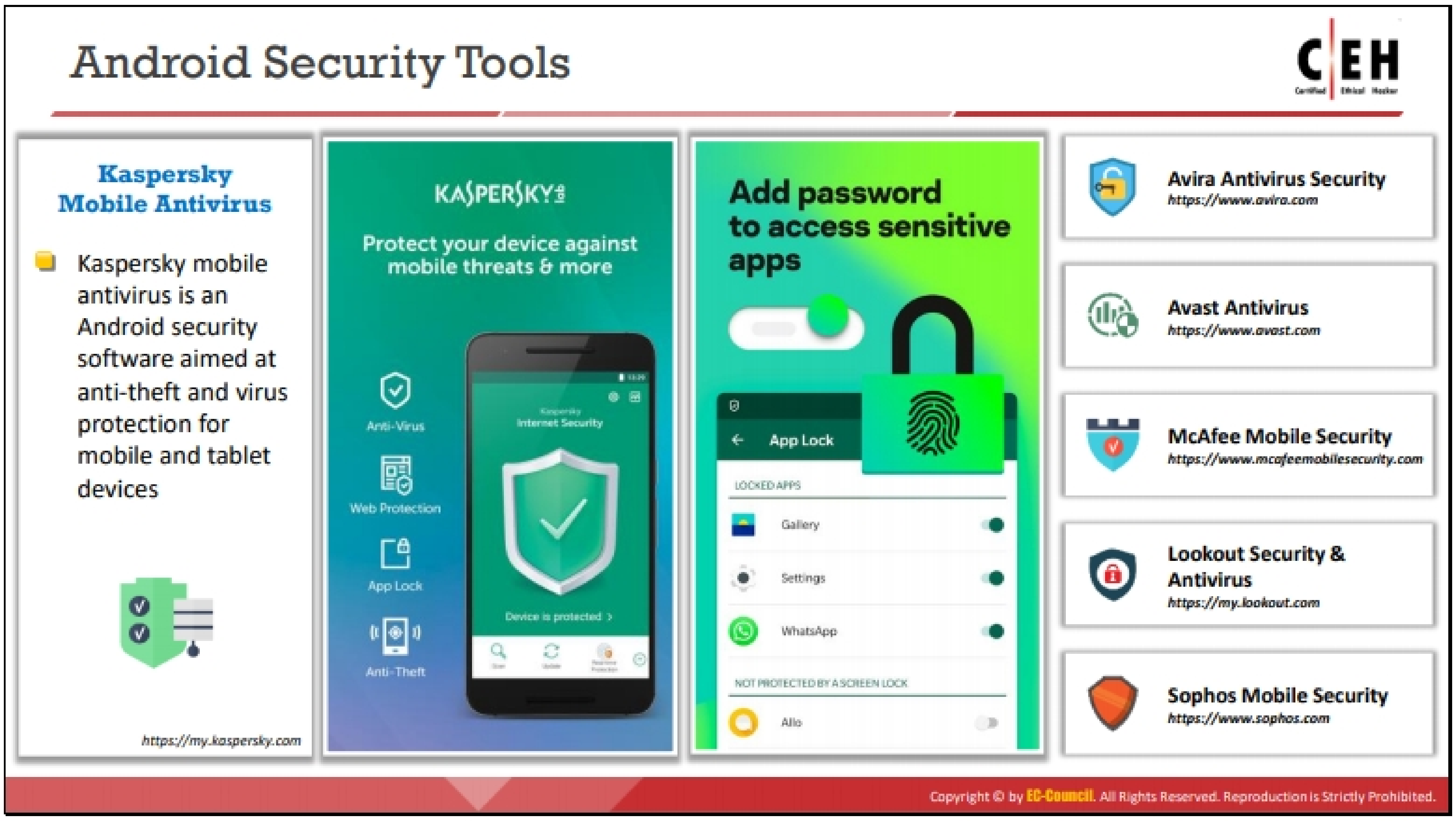
Securing Android Devices

Given below are some countermeasures that can help you to protect your Android device and the data stored on it from malicious users:

- Enable screen lock for your Android phone
- Never root your Android device
- Download apps only from official Android markets
- Keep your device updated with Google Android anti-virus software
- Do not directly download APKs
- Update the OS regularly
- Use free protector Android apps such as Android Protector, where you can assign passwords to text messages, mail accounts, and so on
- Customize your locked home screen with the user information
- Enable encryption in your Android device to enhance its security
- Lock your apps that hold private information to prevent others from viewing it, using apps such as AppLock
- Prior to installing an app from Google Play, read the required permissions and ensure it makes sense and corresponds to what the app actually does, and go through the comments and rating of that app
- Create multiple accounts if you would like to share your Android tablet with others, to protect each user's privacy

- Enable GPS on your Android device for it to be tracked when lost or stolen
- Use third-party applications such as Lookout Mobile Security, 3CX Mobile Device Manager, or SeekDroid to remotely wipe confidential data on your Android device when it is lost or stolen
- Turn off the following features:
    - “**Visible Passwords**”—prevents displaying passwords on screen
    - “**Use Secure Credentials**”—prevents applications from accessing secure certificates and credentials
    - “**Wi-Fi**”—to ensure that you do not inadvertently connect to a wireless network when you do not wish to
- Uninstall apps that invade your privacy
- Encrypt all the Internet traffic through VPN services such as ExpressVPN and VyprVPN for Android.
- Block all the ads displayed by apps
- Enable two-step verification on your Android mobile device
- Disable features such as SmartLock instead of passwords, and auto sign-in functionality.
- Install password manager apps such as LastPass to manage passwords securely
- Enable the screen pinning option to securely access Android apps

Note: You can find many of the features discussed above in **Settings** → **Connections** and in **Settings** → **More** → **Security** on most Android devices.



Android Security Tools

Kaspersky Mobile Antivirus

- Kaspersky mobile antivirus is an Android security software aimed at anti-theft and virus protection for mobile and tablet devices

https://my.kaspersky.com

KASPERSKY
Protect your device against mobile threats & more
Anti-Virus
Web Protection
App Lock
Anti-Theft
Kaspersky Internet Security
Device is protected
Add password to access sensitive apps
App Lock
LOCKED APPS
Gallery
Settings
WhatsApp
NOT PROTECTED BY A SCREEN LOCK
Allo

Avira Antivirus Security
https://www.avira.com

Avast Antivirus
https://www.avast.com

McAfee Mobile Security
https://www.mcafeemobilesecurity.com

Lookout Security & Antivirus
https://my.lookout.com

Sophos Mobile Security
https://www.sophos.com

Android Security Tools

- **Kaspersky Mobile Antivirus**

  Source: *https://my.kaspersky.com*

  Kaspersky Mobile Antivirus is an Android security app focusing on anti-theft and virus protection for mobile devices and tablets. It is designed to help users find their device if it is lost or stolen. It also protects the device against viruses or malware attacks. It provides features such as anti-virus protection, background check, app lock, anti-theft, and anti-phishing.

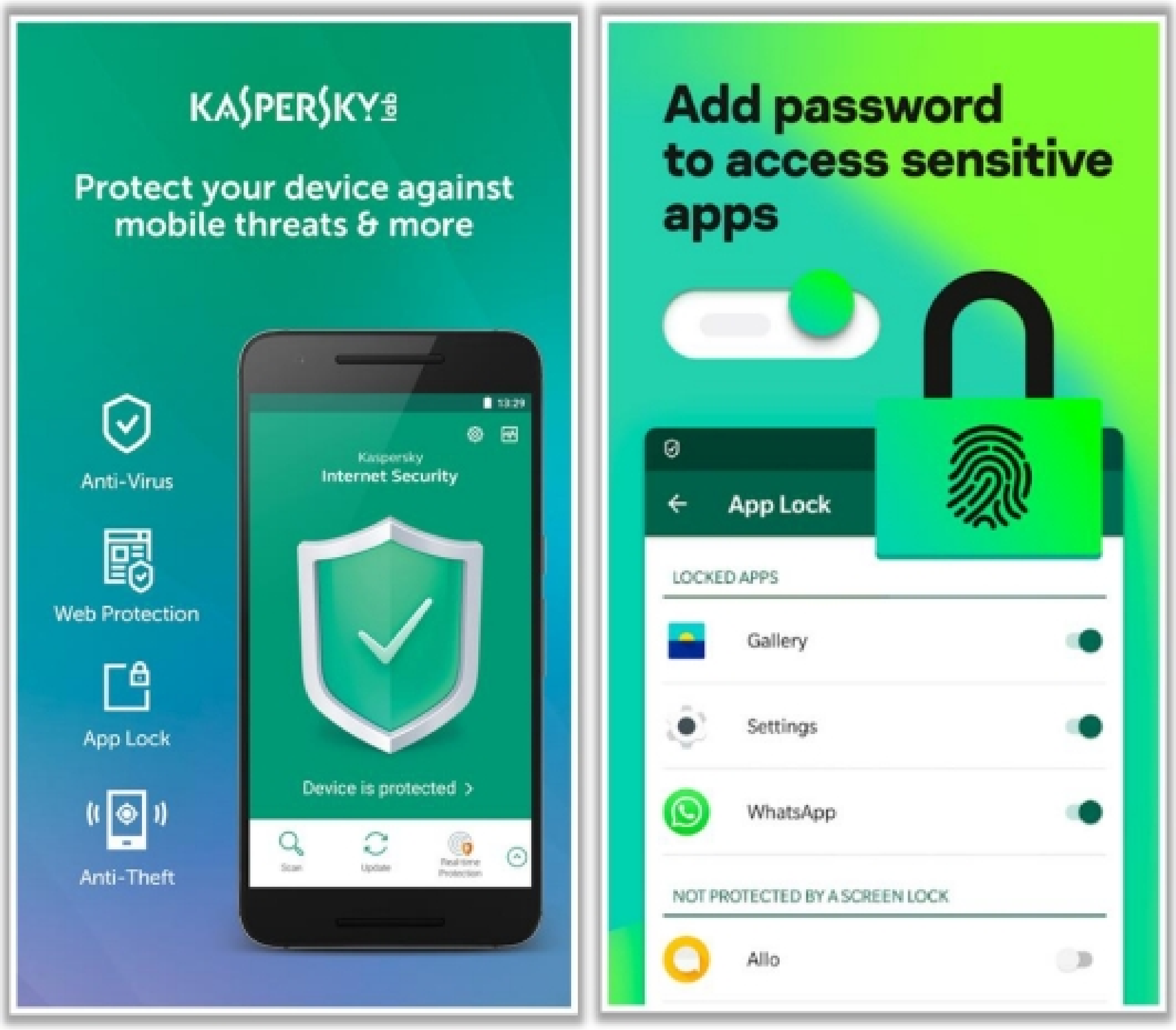

KASPERSKY lab
Protect your device against mobile threats & more
Anti-Virus
Web Protection
App Lock
Anti-Theft
Kaspersky Internet Security
Device is protected
Scan
Update
Real-time Protection
Add password to access sensitive apps
App Lock
LOCKED APPS
Gallery
Settings
WhatsApp
NOT PROTECTED BY A SCREEN LOCK
Allo

Figure 17.36: Screenshot of Kaspersky Mobile Antivirus

Some additional Android security tools are as follows:

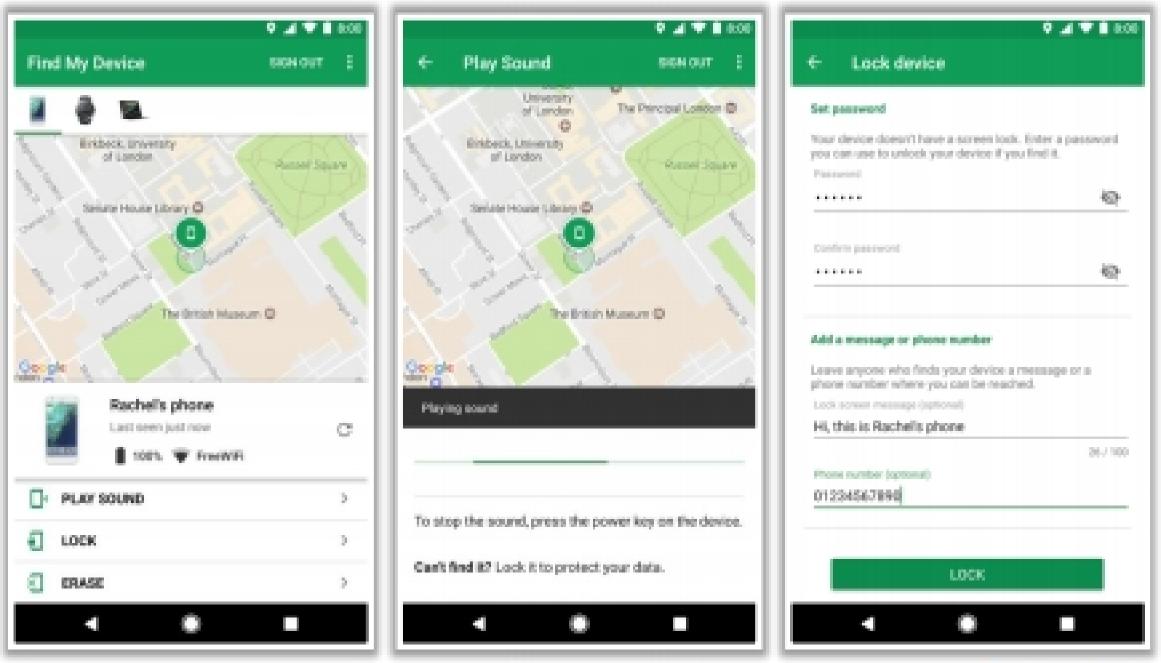
- Avira Antivirus Security (*https://www.avira.com*)
- Avast Antivirus (*https://www.avast.com*)
- McAfee Mobile Security (*https://www.mcafeemobilesecurity.com*)
- Lookout Security & Antivirus (*https://my.lookout.com*)
- Sophos Mobile Security (*https://www.sophos.com*)

Android Device Tracking Tools: Google Find My Device

- Google's Find My Device helps you easily **locate a lost Android device** and keeps your information on the missing device safe while you look for it

To find, lock, or erase a lost or stolen device:

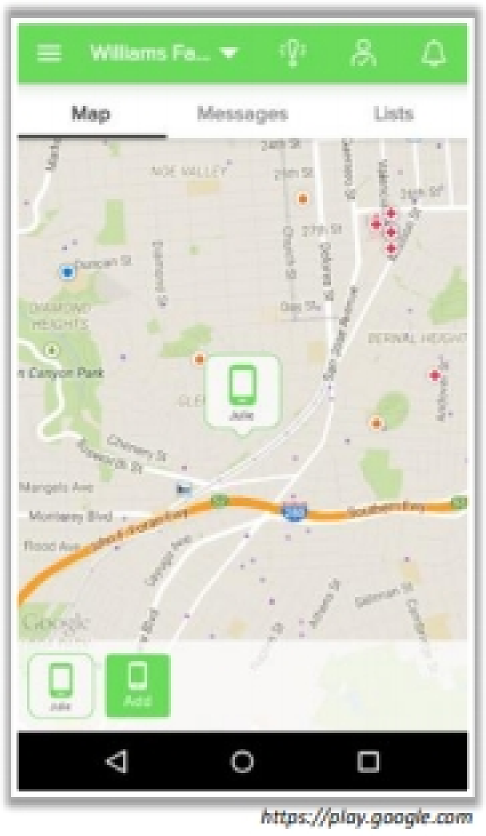
- Go to https://www.google.com/android/find and sign in to your Google Account
- If you have more than one device, click "**Lost device**" at the top of the screen
- The device gets a **notification**
- Locate the device on the map
- Pick what you want to do. If needed, first click "**Enable lock & erase**"
  - **Play sound**: Rings your device at full volume for 5 minutes
  - **Secure Device**: Locks your device with your PIN, pattern, or password
  - **Erase Device**: Permanently deletes all data on your device


Find My Device
SIGN OUT
Rachel's phone
PLAY SOUND
LOCK
ERASE
Play Sound
Playing sound
To stop the sound, press the power key on the device.
Can't find it? Lock it to protect your data.
Lock device
Set password
Add a message or phone number
Hi, this is Rachel's phone
01234567890
LOCK

https://www.google.com

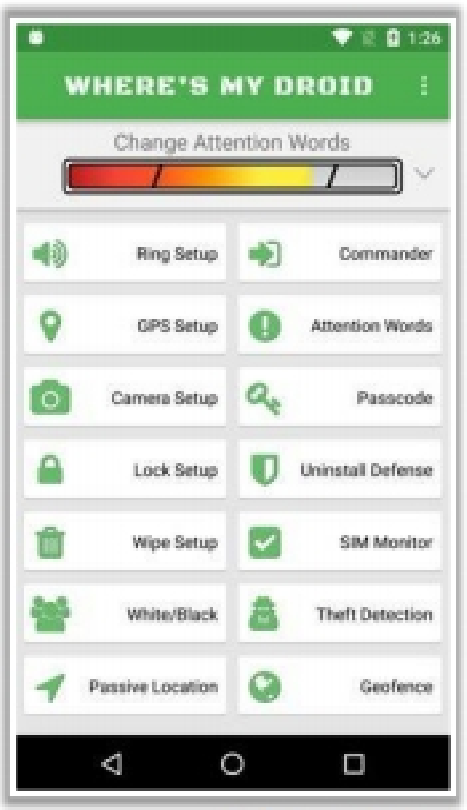
Android Device Tracking Tools

Find My Phone


Williams Fa...
Map
Messages
Lists
Julie
Add

https://play.google.com

Where's My Droid


WHERE'S MY DROID
Change Attention Words
Ring Setup
Commander
GPS Setup
Attention Words
Camera Setup
Passcode
Lock Setup
Uninstall Defense
Wipe Setup
SIM Monitor
White/Black
Theft Detection
Passive Location
Geofence

https://wheresmydroid.com

- **Prey Anti Theft: Find My Android & Mobile Security**
  https://preyproject.com
- **iHound**
  https://www.ihound.com.au
- **Mobile Tracker for Android**
  https://play.google.com
- **Lost Android**
  https://www.androidlost.com
- **Phone Tracker By Number**
  https://play.google.com

Android Device Tracking Tools

Android device tracking tools help you track and find the location of your Android device if it is lost, stolen, or misplaced. Attackers use these tools to track the location of the target mobile devices.

Some widely used Android device tracking tools are listed below:

- **Google Find My Device**

  Source: *https://www.google.com*

  Google Find My Device helps you to easily locate your lost Android device and keeps your information safe in the meantime. It also allows you to erase the information on the lost or stolen device. If users have Google Sync installed on a supported mobile device (including Android) with the Google Apps Device Policy app, they can use the Google Apps control panel to remotely find, lock, or erase a lost Android device.

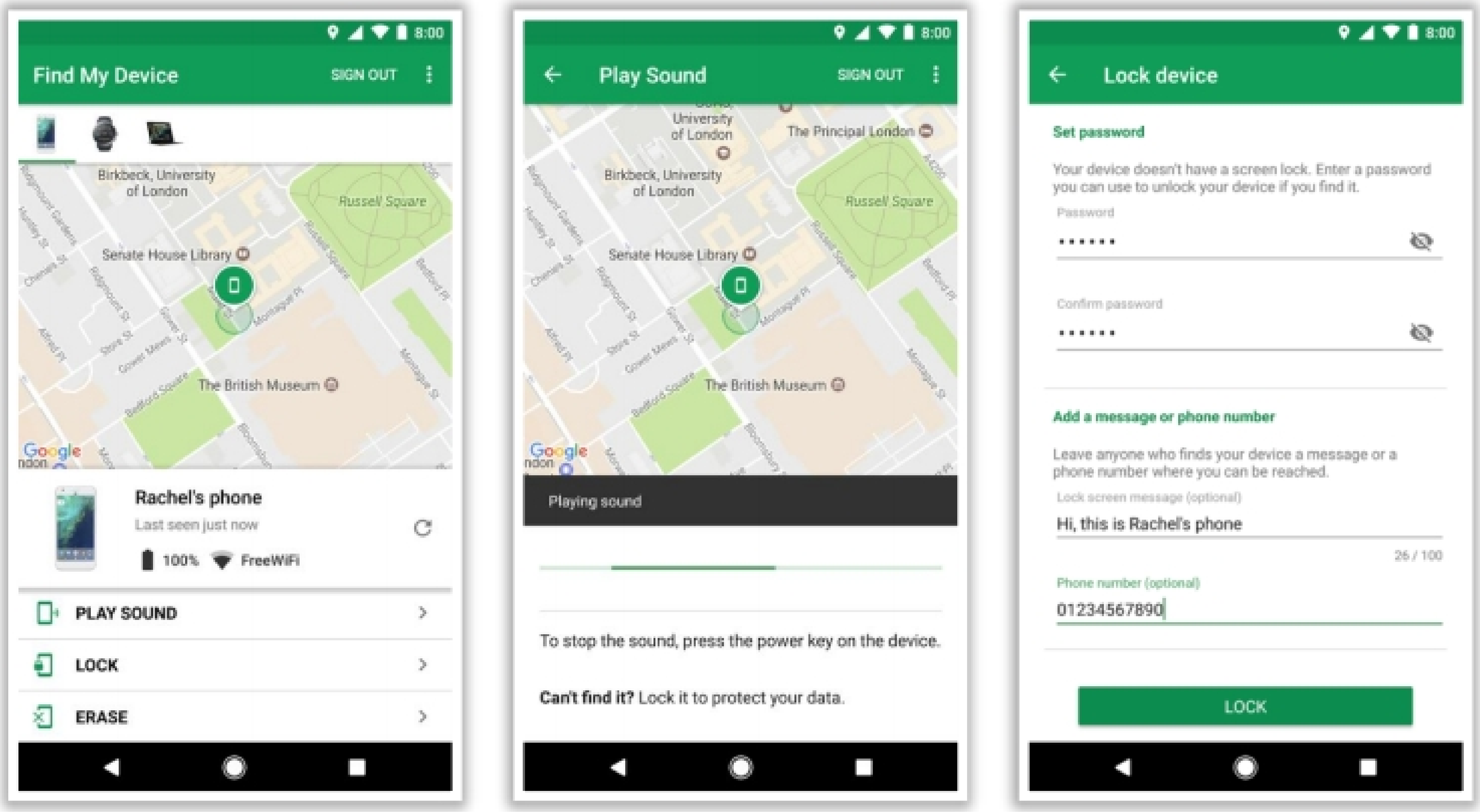
  One can select this service when a device is lost or stolen to erase all the data on the device and perform a factory reset. All the data are erased from the device (and SD card, if applicable), including email, calendar, contacts, photos, music, and the user's personal files.

  To use Find My Device, your lost device must

  - Be turned on
  - Be signed in to a Google Account
  - Be connected to mobile data or Wi-Fi
  - Be visible on Google Play
  - Have Location turned on
  - Have Find My Device turned on

  To find, lock, or erase a lost or stolen device, follow the steps given below:

  - Go to **https://www.google.com/android/find** and sign in to your **Google Account**.
  - If you have more than one device, click the lost device at the top of the screen.
  - The device gets a notification.
  - On the map, see where the device is.
    - The location is approximate and might not be accurate.
    - If your device cannot be found, then you will see its last known location, if available.
  - Pick what you want to do. If needed, first click **Enable lock & erase**.
    - **Play Sound**: Rings your device at full volume for 5 minutes, even if it is set to silent or vibrate.
    - **Secure Device**: Locks your device with your PIN, pattern, or password. If you do not have a lock, you can set one. To enable someone to return your device to you, you can add a message or phone number to the lock screen.
    - **Erase Device**: Permanently deletes all data on your device (but might not delete SD cards). Subsequently, Find My Device will not work on the device.


8:00
Find My Device
SIGN OUT
Rachel's phone
Last seen just now
100%
FreeWiFi
PLAY SOUND
LOCK
ERASE
Play Sound
Playing sound
Lock device
Set password
Password
Confirm password
Add a message or phone number
26 / 100
LOCK
Birkbeck, University of London
Senate House Library
Russell Square
The British Museum
The Principal London
University of London
Google

Your device doesn't have a screen lock. Enter a password you can use to unlock your device if you find it.

Leave anyone who finds your device a message or a phone number where you can be reached.

Lock screen message (optional): Hi, this is Rachel's phone

Phone number (optional): 01234567890

To stop the sound, press the power key on the device.

Can't find it? Lock it to protect your data.

Figure 17.37: Screenshot of Find My Device service

- **Find My Phone**

  Source: *https://play.google.com*

  Find My Phone is an anti-theft device recovery app for Android that helps you find your lost, stolen, or misplaced mobile phone or tablet.

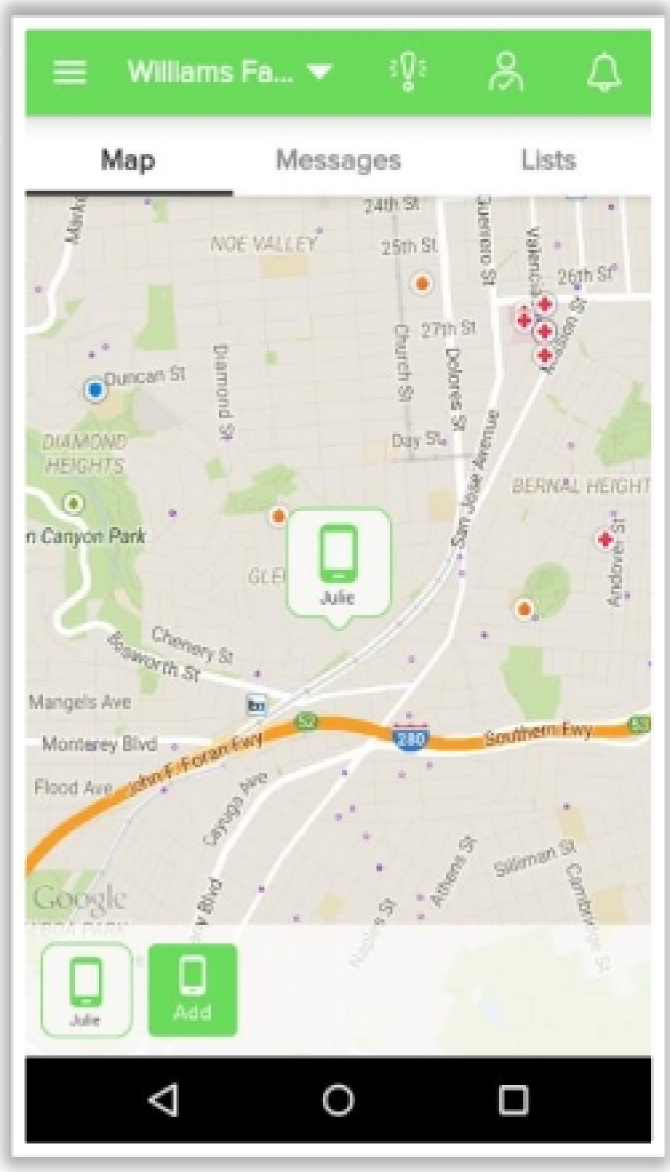

Williams Fa...
Map
Messages
Lists
NOE VALLEY
DIAMOND HEIGHTS
BERNAL HEIGHTS
Julie
Add
Google

Figure 17.38: Screenshot of Find My Phone

- **Where's My Droid**

  Source: *https://wheresmydroid.com*

  Where's My Droid is an Android device tracking tool that allows you to track your phone from anywhere, either with a text-messaged attention word or through the online control center known as Commander.

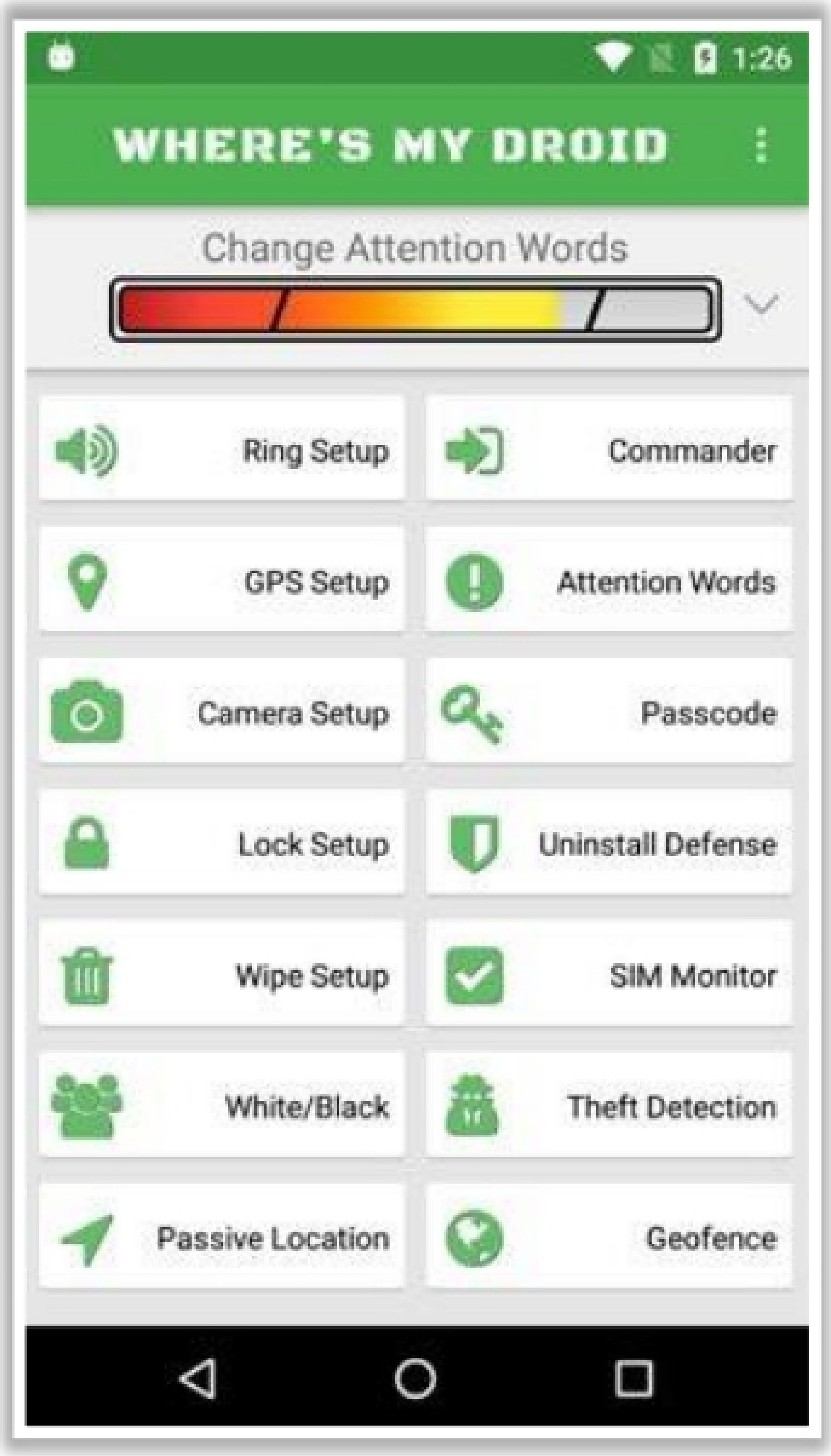
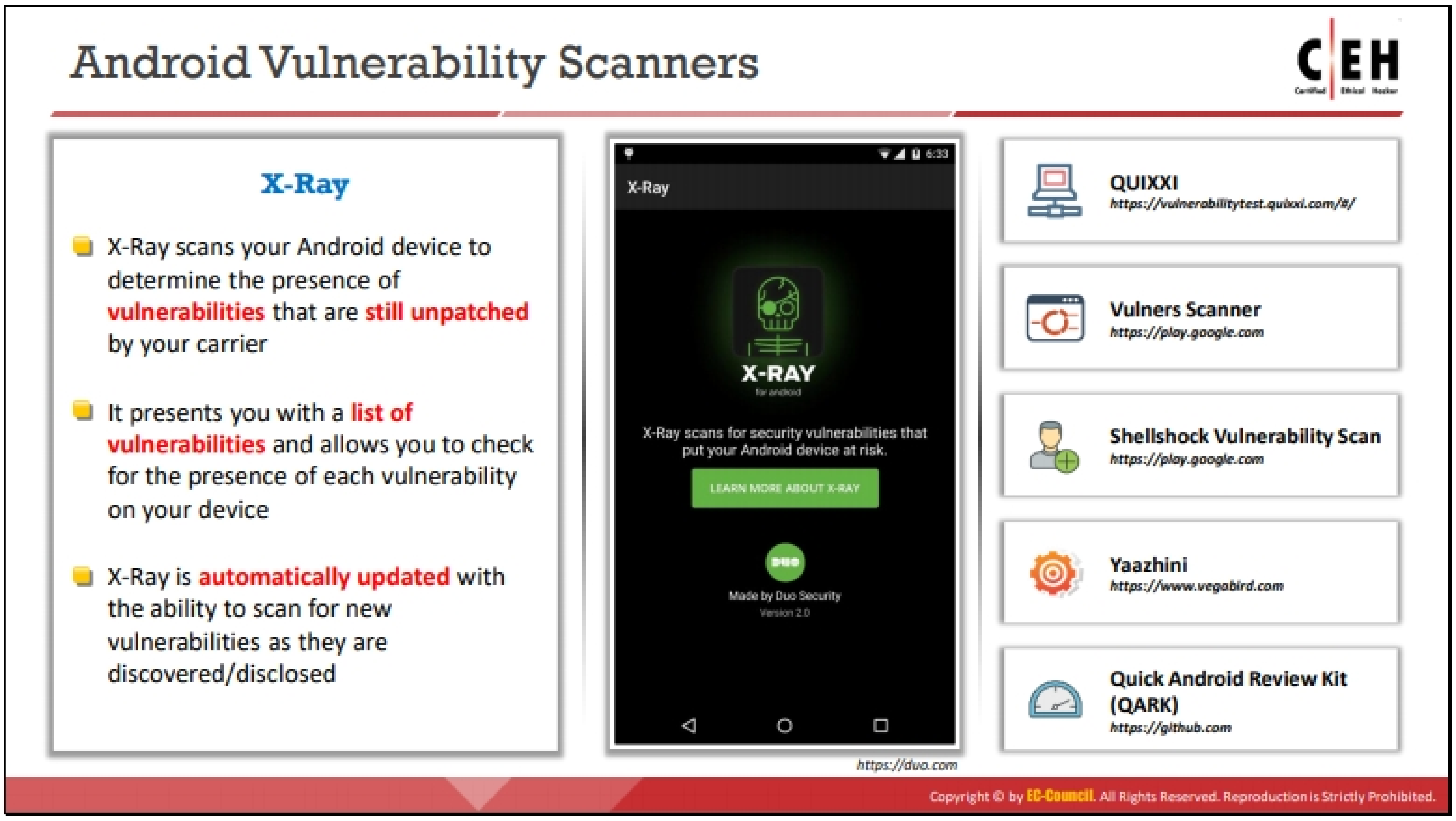

1:26
WHERE'S MY DROID
Change Attention Words
Ring Setup
Commander
GPS Setup
Attention Words
Camera Setup
Passcode
Lock Setup
Uninstall Defense
Wipe Setup
SIM Monitor
White/Black
Theft Detection
Passive Location
Geofence

Figure 17.39: Screenshot of Where's My Droid

Some additional Android device tracking tools are as follows:

- Prey Anti-Theft: Find My Android and Mobile Security (*https://preyproject.com*)
- iHound (*https://www.ihound.com.au*)
- Mobile Tracker for Android (*https://play.google.com*)
- Lost Android (*https://www.androidlost.com*)
- Phone Tracker By Number (*https://play.google.com*)



Android Vulnerability Scanners

X-Ray

- X-Ray scans your Android device to determine the presence of **vulnerabilities** that are **still unpatched** by your carrier
- It presents you with a **list of vulnerabilities** and allows you to check for the presence of each vulnerability on your device
- X-Ray is **automatically updated** with the ability to scan for new vulnerabilities as they are discovered/disclosed

X-Ray scans for security vulnerabilities that put your Android device at risk.

Made by Duo Security
Version 2.0

https://duo.com

QUIXXI
https://vulnerabilitytest.quixxi.com/#/

Vulners Scanner
https://play.google.com

Shellshock Vulnerability Scan
https://play.google.com

Yaazhini
https://www.vegabird.com

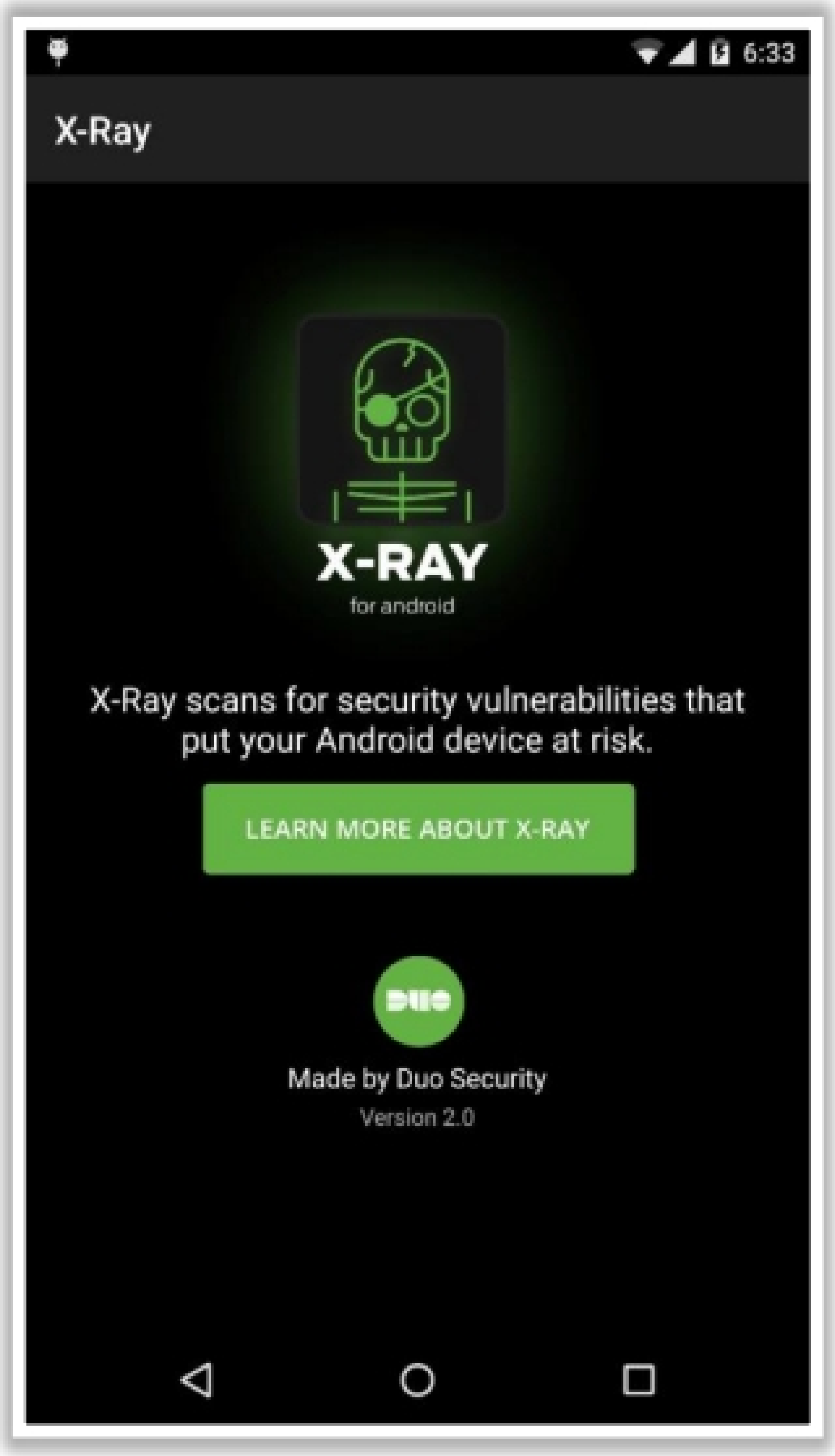
Quick Android Review Kit (QARK)
https://github.com

Android Vulnerability Scanners

- **X-Ray**

  Source: *https://duo.com*

  X-Ray allows you to scan your Android device for security vulnerabilities that put your device at risk. It determines whether there are vulnerabilities that remain unpatched by your carrier. It presents you with a list of vulnerabilities that it can identify and allows you to check for the presence of each vulnerability on your device. X-Ray is automatically updated with the ability to scan for new vulnerabilities as they are discovered and disclosed.



X-Ray scans for security vulnerabilities that put your Android device at risk.

Made by Duo Security
Version 2.0

Figure 17.40: Screenshot of X-Ray Android Vulnerability Scanner

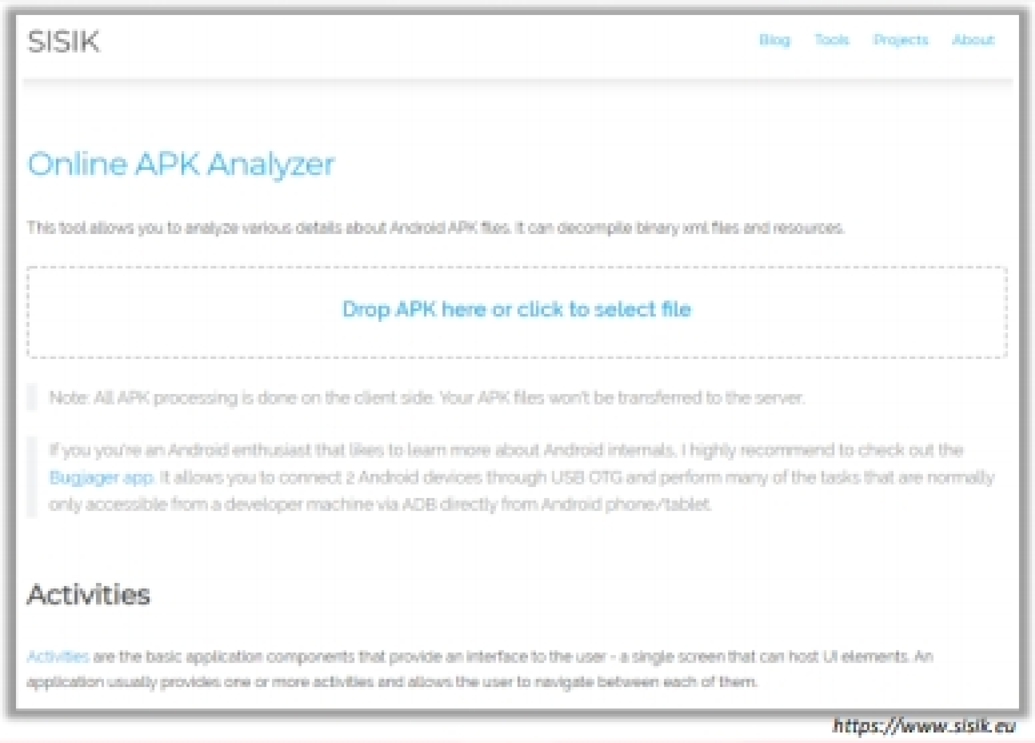
Some additional Android vulnerability scanners are as follows:

- QUIXXI (*https://vulnerabilitytest.quixxi.com/#/*)
- Vulners Scanner (*https://play.google.com*)
- Shellshock Vulnerability Scan (*https://play.google.com*)
- Yaazhini (*https://www.vegabird.com*)
- Quick Android Review Kit (QARK) (*https://github.com*)

Online Android Analyzers

Online APK Analyzer

An online APK Analyzer allows you to analyze various details about Android APK files. It can **decompile binary xml files** and resources



SISIK

Online APK Analyzer

This tool allows you to analyze various details about Android APK files. It can decompile binary xml files and resources.

Note: All APK processing is done on the client side. Your APK files won't be transferred to the server.

If you you're an Android enthusiast that likes to learn more about Android internals, I highly recommend to check out the Bugjaeger app. It allows you to connect 2 Android devices through USB OTG and perform many of the tasks that are normally only accessible from a developer machine via ADB directly from Android phone/tablet.

Activities

Activities are the basic application components that provide an interface to the user - a single screen that can host UI elements. An application usually provides one or more activities and allows the user to navigate between each of them.

https://www.sisik.eu

- **DeGuard** *http://apk-deguard.com*
- **SandDroid** *http://sanddroid.xjtu.edu.cn*
- **Apktool** *http://www.javadecompilers.com*
- **UnDroid** *https://undroid.av-comparatives.org*
- **Apprisk Scanner** *https://apprisk.newskysecurity.com*

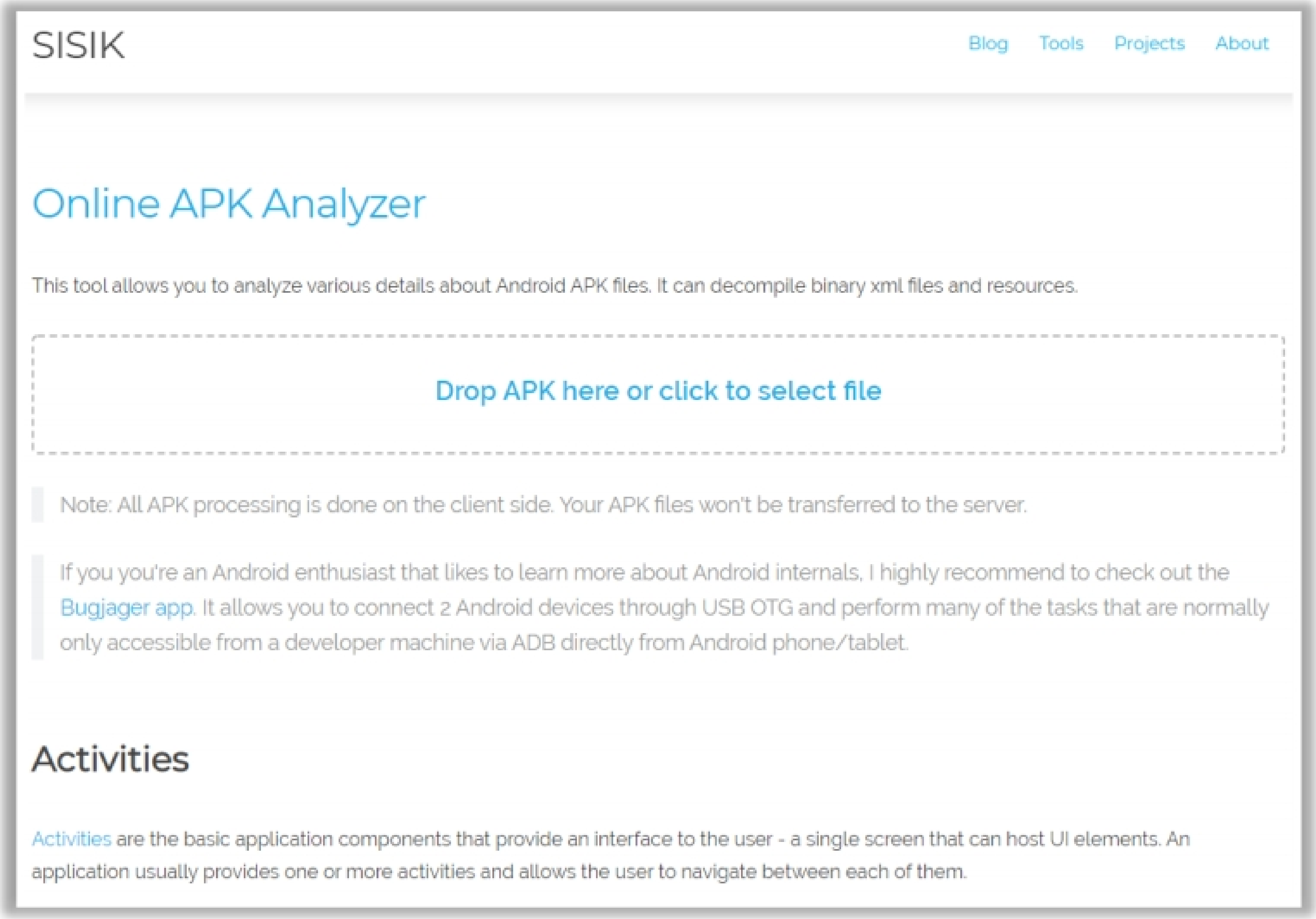
Online Android Analyzers

Online Android analyzers allow you to scan APKs and perform security analysis to detect vulnerabilities in the applications.

- **Online APK Analyzer**

  Source: *https://www.sisik.eu*

  Online APK Analyzer allows you to analyze various details about APK files. It can decompile binary XML files and resources.



Online APK Analyzer

This tool allows you to analyze various details about Android APK files. It can decompile binary xml files and resources.

Note: All APK processing is done on the client side. Your APK files won't be transferred to the server.

If you you're an Android enthusiast that likes to learn more about Android internals, I highly recommend to check out the Bugjager app. It allows you to connect 2 Android devices through USB OTG and perform many of the tasks that are normally only accessible from a developer machine via ADB directly from Android phone/tablet.

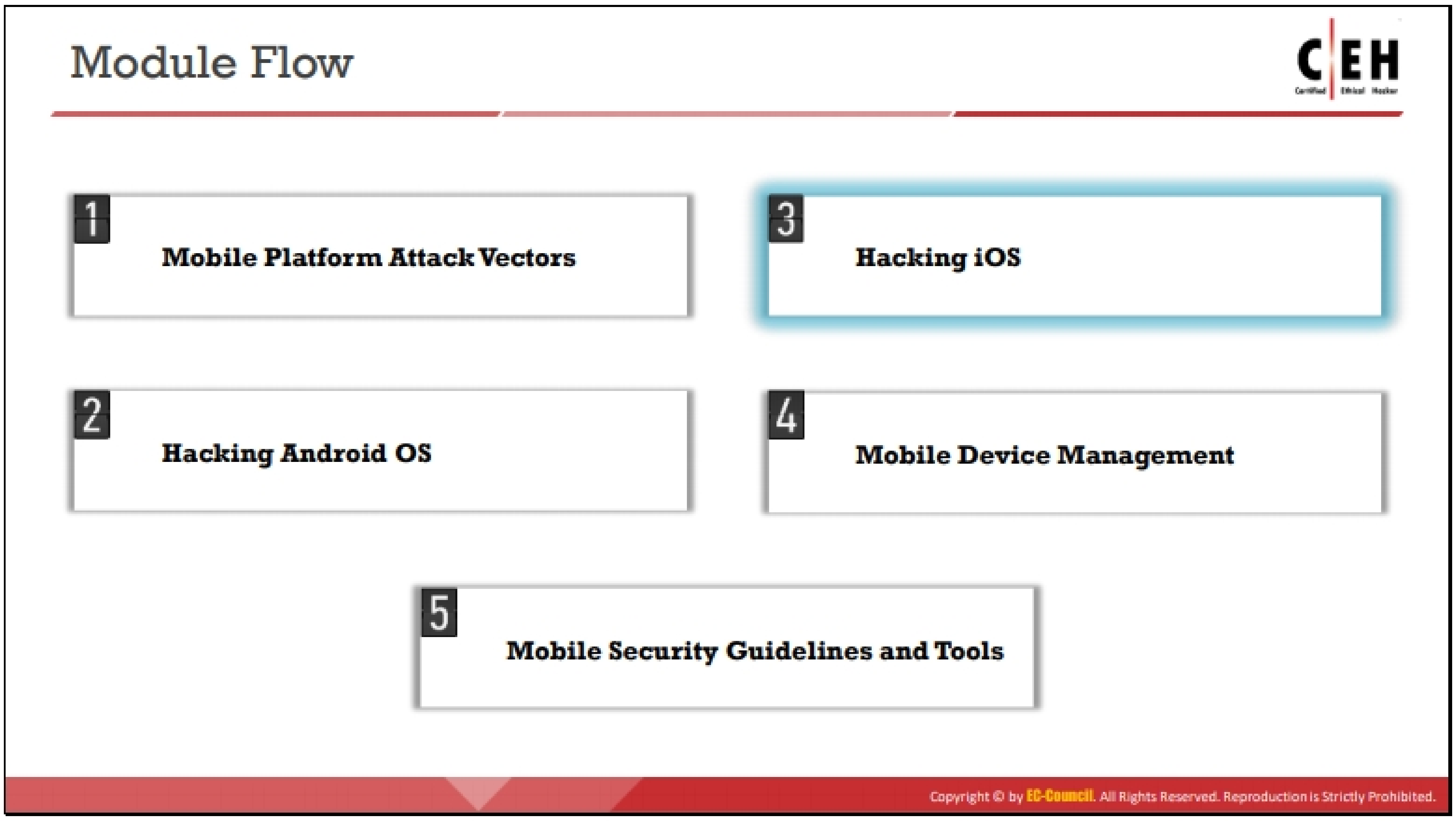
Activities

Activities are the basic application components that provide an interface to the user - a single screen that can host UI elements. An application usually provides one or more activities and allows the user to navigate between each of them.

Figure 17.41: Screenshot of Online APK Analyzer

Some additional online Android analyzers are as follows:

- DeGuard (*http://apk-deguard.com*)
- SandDroid (*http://sanddroid.xjtu.edu.cn*)
- Apktool (*http://www.javadecompilers.com*)
- UnDroid (*https://undroid.av-comparatives.org*)
- Apprisk Scanner (*https://apprisk.newskysecurity.com*)



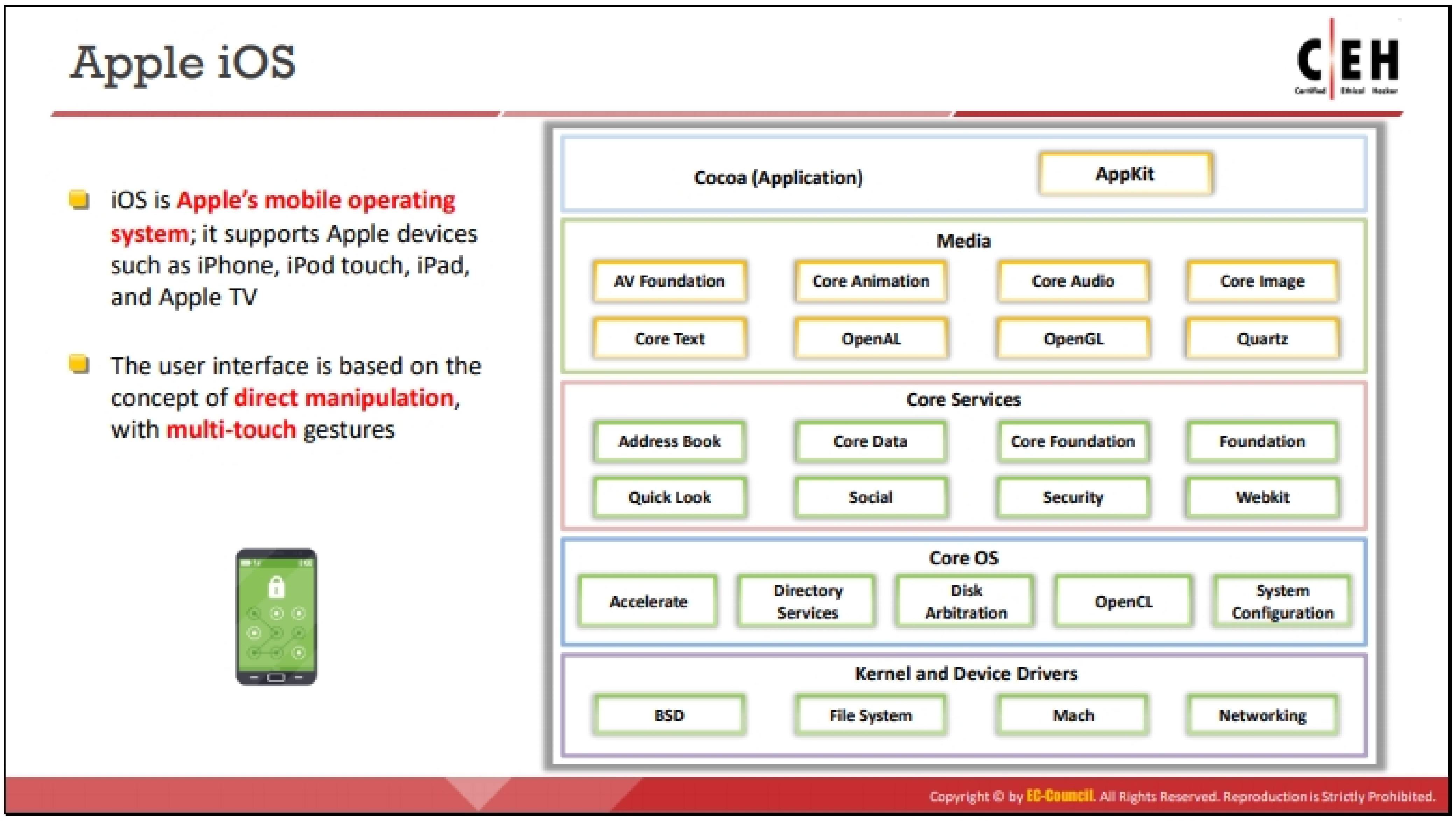
Module Flow

1. Mobile Platform Attack Vectors
2. Hacking Android OS
3. Hacking iOS
4. Mobile Device Management
5. Mobile Security Guidelines and Tools

Hacking iOS

iOS is a mobile OS developed by Apple. Apple does not license iOS for installation on non-Apple hardware. The company has increased its product range by including mobile phones, tablets, and other mobile devices. The rapid increase in the use of Apple devices has attracted the attention of attackers. The design flaws in iOS make it vulnerable to malicious apps, hidden network profiles, MITM attacks, etc. Attackers can hack iOS to gain root-level access to Apple devices.

This section introduces the following: Apple iOS; jailbreaking iOS; types, tools, and techniques of jailbreaking; guidelines for securing iOS devices; and iOS device tracking tools.



Apple iOS

- iOS is **Apple's mobile operating system**; it supports Apple devices such as iPhone, iPod touch, iPad, and Apple TV
- The user interface is based on the concept of **direct manipulation**, with **multi-touch** gestures

Cocoa (Application)
AppKit

Media
AV Foundation
Core Animation
Core Audio
Core Image
Core Text
OpenAL
OpenGL
Quartz

Core Services
Address Book
Core Data
Core Foundation
Foundation
Quick Look
Social
Security
Webkit

Core OS
Accelerate
Directory Services
Disk Arbitration
OpenCL
System Configuration

Kernel and Device Drivers
BSD
File System
Mach
Networking

Apple iOS

iOS is Apple's mobile OS, which supports Apple devices such as iPhone, iPod touch, iPad, and Apple TV. iOS manages the device hardware and offers various technologies required to implement native apps. At the highest level, iOS acts as an intermediary between apps and the underlying hardware. Apps communicate with the underlying hardware via a set of well-defined system interfaces. The UI is based on the concept of direct manipulation, using multi-touch gestures. The iOS architecture comprises five layers: Cocoa application, media, core services, core OS and kernel, and device drivers. The lower-level layers contain fundamental services and technologies, whereas the higher-level layers build upon the lower layers to provide more sophisticated services and technologies.

- **Cocoa Application**: This layer contains key frameworks that help in building iOS apps. These frameworks define the appearance of the apps, offer basic app infrastructure, and support key technologies such as multitasking, touch-based input, push notifications, and many high-level system services. Cocoa apps use the AppKit framework.
- **Media**: This layer contains the graphics, audio, and video technologies that enable multimedia experiences in apps.
- **Core Services**: This layer contains fundamental system services for apps. The key services are Core Foundation and Foundation frameworks (define the basic types that all apps use). Individual technologies that support features such as social media, iCloud, location, and networking belong to this layer.
- **Core OS**: This layer contains low-level features on which most other technologies are based. Frameworks in this layer are useful when dealing explicitly with security or

communicating with an external hardware and networks. The services provided by this layer are dependent on the Kernel and Device Drivers layer.

- **Kernel and Device Drivers**: The lowest layer of the iOS architecture includes the kernel, drivers, BSD, file systems, infrastructure technologies such as networking.

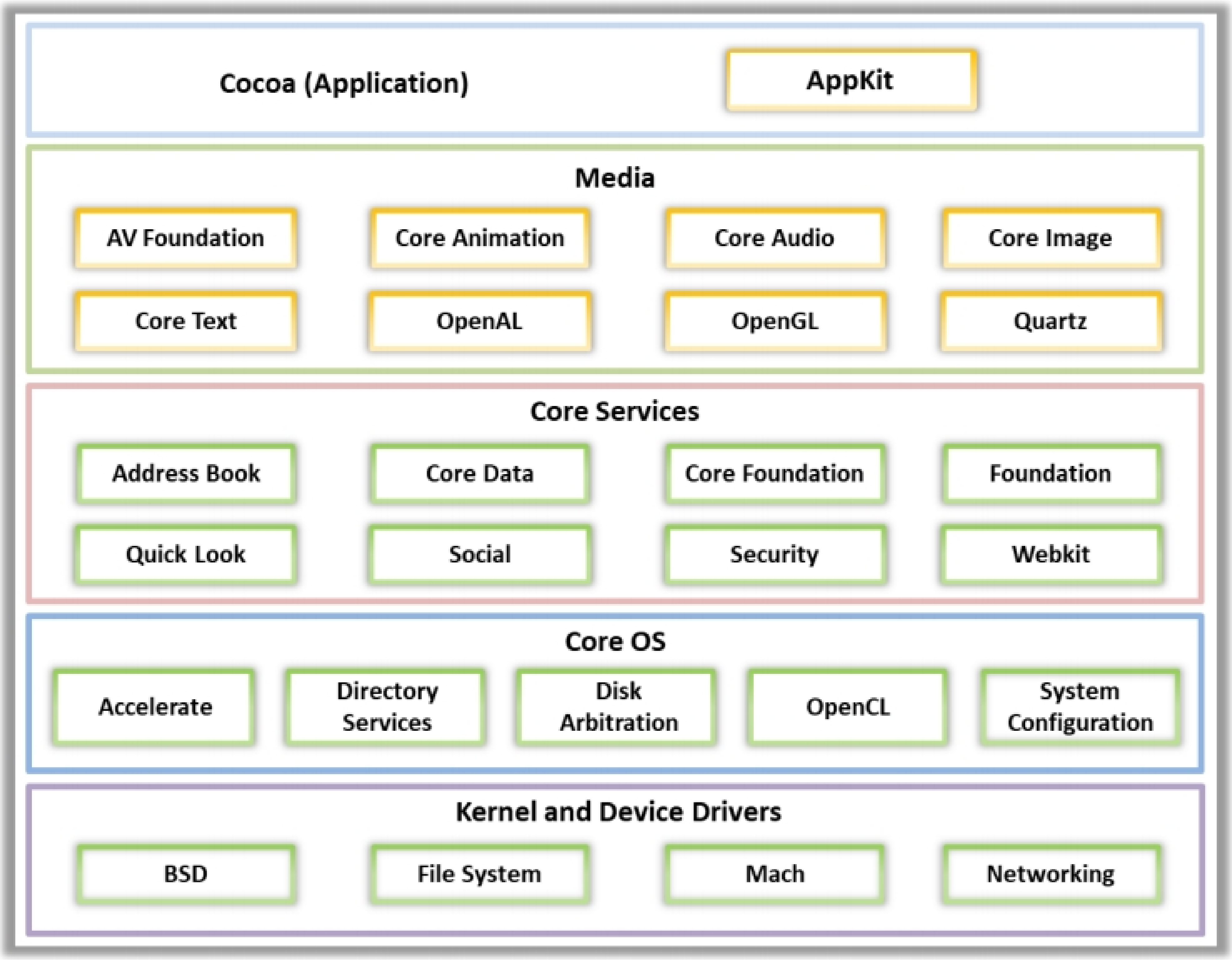
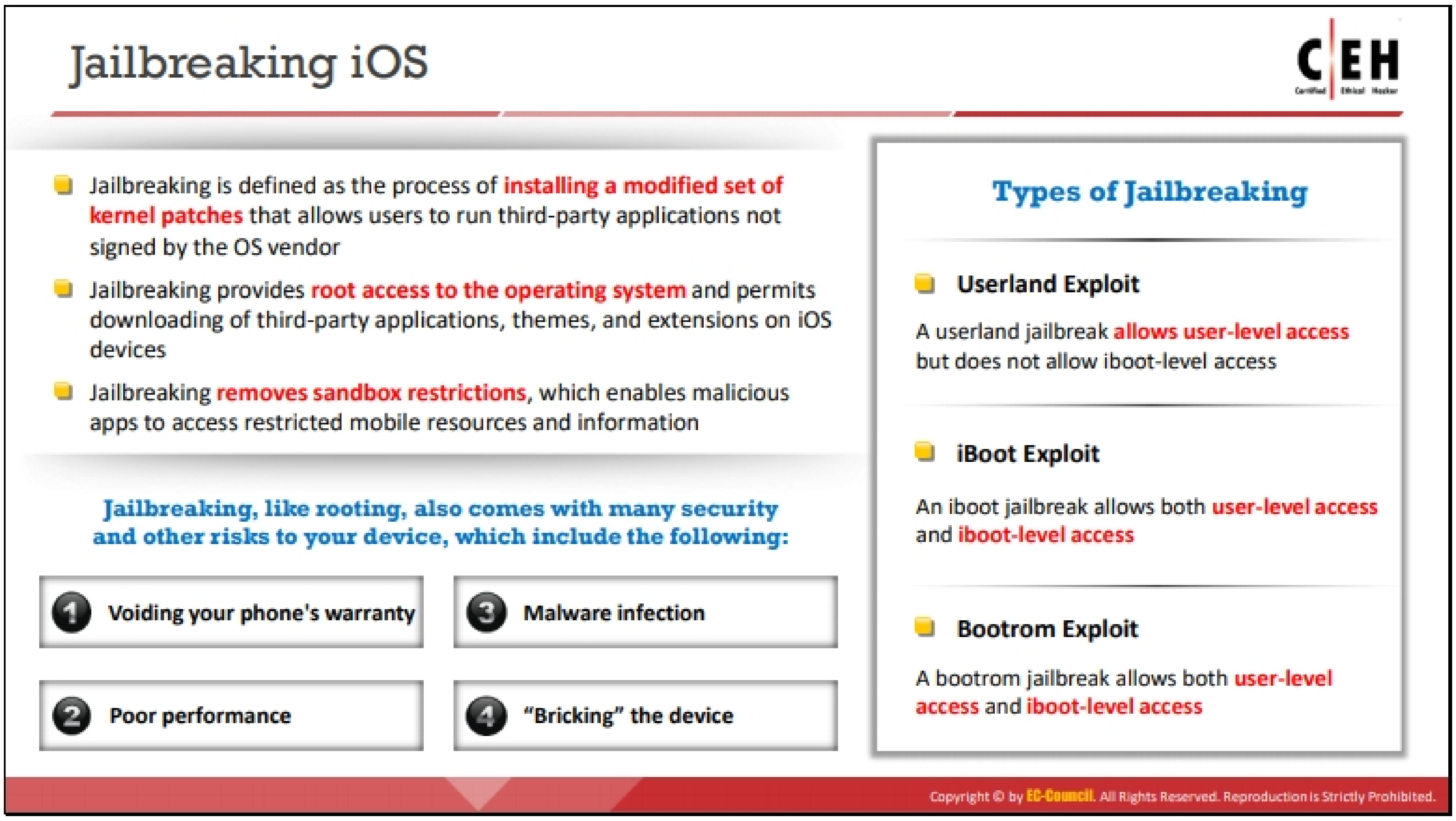

Cocoa (Application)
AppKit
Media
AV Foundation
Core Animation
Core Audio
Core Image
Core Text
OpenAL
OpenGL
Quartz
Core Services
Address Book
Core Data
Core Foundation
Foundation
Quick Look
Social
Security
Webkit
Core OS
Accelerate
Directory Services
Disk Arbitration
OpenCL
System Configuration
Kernel and Device Drivers
BSD
File System
Mach
Networking

Figure 17.42: iOS Framework



Jailbreaking iOS

- Jailbreaking is defined as the process of **installing a modified set of kernel patches** that allows users to run third-party applications not signed by the OS vendor
- Jailbreaking provides **root access to the operating system** and permits downloading of third-party applications, themes, and extensions on iOS devices
- Jailbreaking **removes sandbox restrictions**, which enables malicious apps to access restricted mobile resources and information

Jailbreaking, like rooting, also comes with many security and other risks to your device, which include the following:

1. **Voiding your phone's warranty**
2. **Poor performance**
3. **Malware infection**
4. **"Bricking" the device**

Types of Jailbreaking

- **Userland Exploit**

  A userland jailbreak **allows user-level access** but does not allow iboot-level access

- **iBoot Exploit**

  An iboot jailbreak allows both **user-level access** and **iboot-level access**

- **Bootrom Exploit**

  A bootrom jailbreak allows both **user-level access** and **iboot-level access**

Jailbreaking iOS

Jailbreaking is defined as the process of installing a modified set of kernel patches that allow users to run third-party applications not signed by the OS vendor. It is the process of bypassing the user limitations set by Apple, such as modifying the OS, attaining admin privileges, and installing unofficially approved apps via "side loading." You can accomplish jailbreaking by simply modifying the iOS system kernels. One reason for jailbreaking iOS devices such as iPhone, iPad, and iPod Touch is to expand the feature set restricted by Apple and its App Store. Jailbreaking provides root access to the OS and permits downloading of third-party applications, themes, and extensions that are unavailable through the official Apple App Store. Jailbreaking also removes sandbox restrictions, allowing malicious apps to access restricted mobile resources and information. One can use tools such as Cydia, Hexxa Plus, ApricotiOS, Yuxigon, Sileo, Trimgo, and so on to jailbreak iOS devices.

Jailbreaking, like rooting, comes with many security risks and other risks to your device, including

- Voiding your phone's warranty
- Poor performance
- Malware infection
- "Bricking" the device

Types of Jailbreaking

The three types of jailbreaking are discussed below:

- **Userland Exploit**

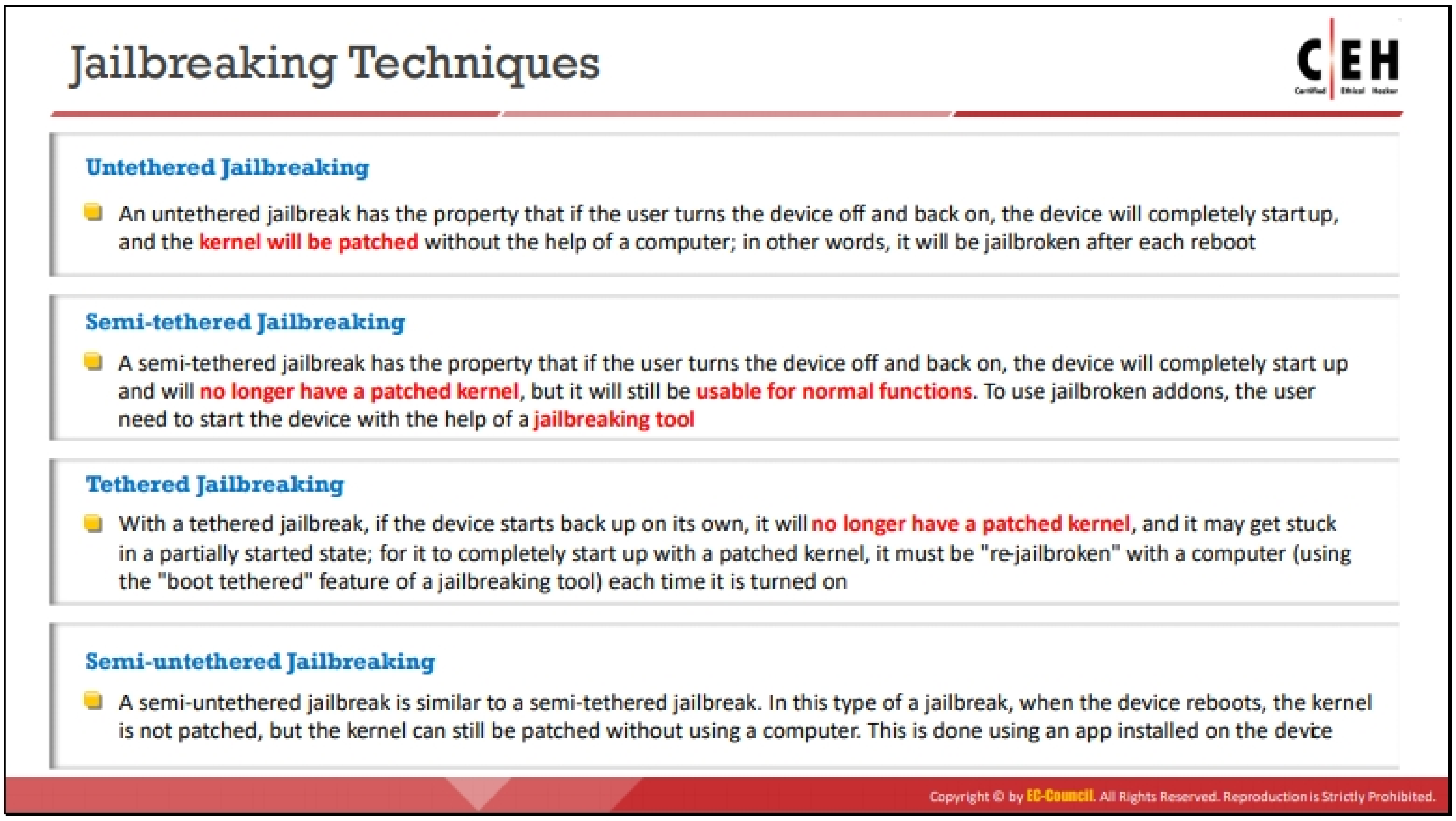
  Userland Exploit uses a loophole in the system application. It allows user-level access but does not allow iboot-level access. You cannot secure iOS devices against this exploit, as nothing can cause a recovery mode loop. Only firmware updates can patch such vulnerabilities.

- **iBoot Exploit**

  This type of exploit can be semi-tethered if the device has a new bootrom. An iboot jailbreak allows user-level access and iboot-level access. This exploit takes advantage of a loophole in iBoot (iDevice's third bootloader) to delink the code-signing appliance. Firmware updates can patch such exploits.

- **Bootrom Exploit**

  Bootrom Exploit uses a loophole in the SecureROM (iDevice's first bootloader) to disable signature checks, which can be used to load patch NOR firmware. Firmware updates cannot patch such exploits. A bootrom jailbreak allows user-level access and iboot-level access. Only a hardware update of bootrom by Apple can patch this exploit.



Jailbreaking Techniques

Untethered Jailbreaking

- An untethered jailbreak has the property that if the user turns the device off and back on, the device will completely start up, and the kernel will be patched without the help of a computer; in other words, it will be jailbroken after each reboot

Semi-tethered Jailbreaking

- A semi-tethered jailbreak has the property that if the user turns the device off and back on, the device will completely start up and will no longer have a patched kernel, but it will still be usable for normal functions. To use jailbroken addons, the user need to start the device with the help of a jailbreaking tool

Tethered Jailbreaking

- With a tethered jailbreak, if the device starts back up on its own, it will no longer have a patched kernel, and it may get stuck in a partially started state; for it to completely start up with a patched kernel, it must be "re-jailbroken" with a computer (using the "boot tethered" feature of a jailbreaking tool) each time it is turned on

Semi-untethered Jailbreaking

- A semi-untethered jailbreak is similar to a semi-tethered jailbreak. In this type of a jailbreak, when the device reboots, the kernel is not patched, but the kernel can still be patched without using a computer. This is done using an app installed on the device

Jailbreaking Techniques

- **Untethered Jailbreaking**

  In an untethered jailbreak, if the user turns the device off and back on, the device will start up completely and the kernel will be patched without the help of a computer; in other words, the device will be jailbroken after each reboot.

- **Semi-tethered Jailbreaking**

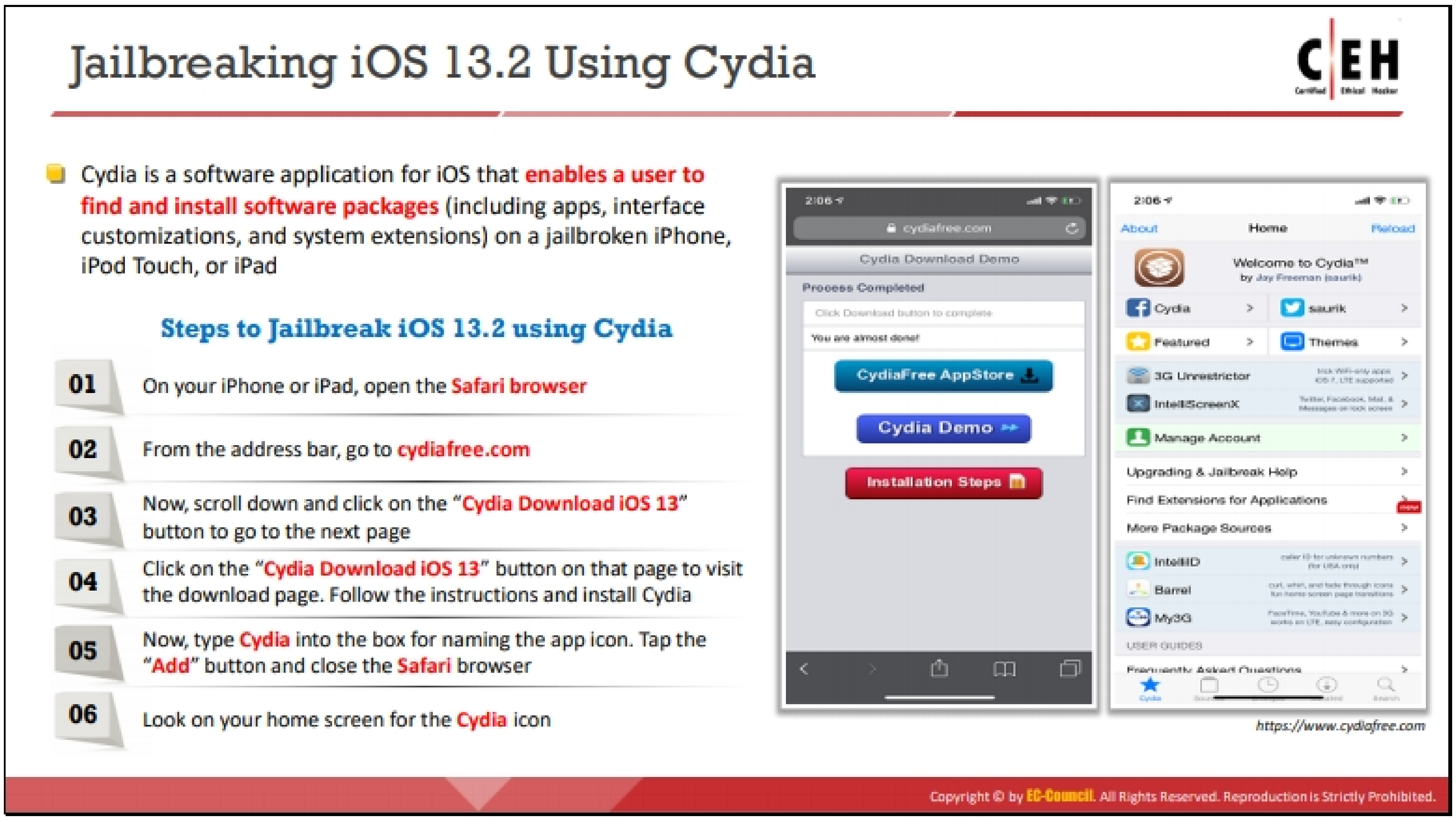
  In a semi-tethered jailbreak, if the user turns the device off and back on, the device will start up completely. It will no longer have a patched kernel, but it will still be usable for normal functions. To use jailbroken addons, the user needs to start the device with the help of the jailbreaking tool.

- **Tethered Jailbreaking**

  With a tethered jailbreak, if the device starts up on its own, it will no longer have a patched kernel, and it may get stuck in a partially started state; to start it completely and with a patched kernel, it essentially must be “re-jailbroken” with a computer (using the “boot tethered” feature of a jailbreaking tool) each time it is turned on.

- **Semi-untethered Jailbreaking**

  A semi-untethered jailbreak is similar to a semi-tethered jailbreak. In this type of jailbreak, when the device reboots, the kernel is not patched. However, the kernel can be patched without using a computer; it is patched using an app installed on the device.



Jailbreaking iOS 13.2 Using Cydia

- Cydia is a software application for iOS that **enables a user to find and install software packages** (including apps, interface customizations, and system extensions) on a jailbroken iPhone, iPod Touch, or iPad

Steps to Jailbreak iOS 13.2 using Cydia

01 On your iPhone or iPad, open the **Safari browser**

02 From the address bar, go to **cydiafree.com**

03 Now, scroll down and click on the "**Cydia Download iOS 13**" button to go to the next page

04 Click on the "**Cydia Download iOS 13**" button on that page to visit the download page. Follow the instructions and install Cydia

05 Now, type **Cydia** into the box for naming the app icon. Tap the "**Add**" button and close the **Safari** browser

06 Look on your home screen for the **Cydia** icon

cydiafree.com
Cydia Download Demo
Process Completed
Click Download button to complete
You are almost done!
CydiaFree AppStore
Cydia Demo
Installation Steps
About
Home
Reload
Welcome to Cydia™
by Jay Freeman (saurik)
Cydia
saurik
Featured
Themes
3G Unrestrictor
IntelliScreenX
Manage Account
Upgrading & Jailbreak Help
Find Extensions for Applications
More Package Sources
IntelliID
Barrel
My3G
USER GUIDES
Frequently Asked Questions

https://www.cydiafree.com

Jailbreaking iOS 13.2 Using Cydia

Source: *https://www.cydiafree.com*

Cydia is a software application for iOS that enables a user to find and install software packages (including apps, interface customizations, and system extensions) on a jailbroken iPhone, iPod Touch, or iPad. It is a graphical front end to the Advanced Packaging Tool (APT) and the dpkg package management system, which means that the packages available in Cydia are provided by a decentralized system of repositories (also called sources) that list these packages.

You can jailbreak iOS 13.2 using Cydia in the following steps:

- On your iPhone or iPad, open the **Safari browser**

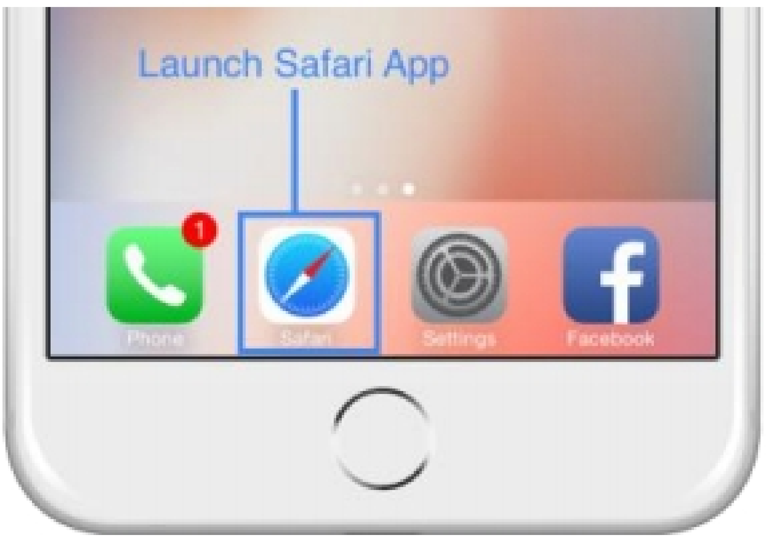

Launch Safari App
1
Phone
Safari
Settings
Facebook

Figure 17.43: Launch Safari App

- From the address bar, go to ***cydiafree.com***

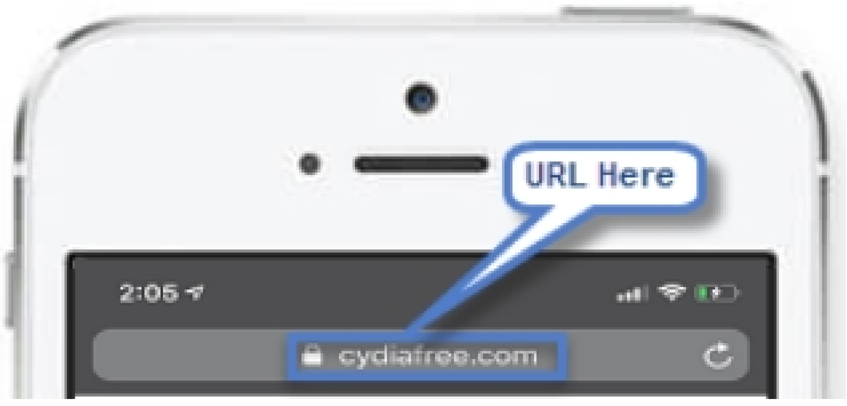
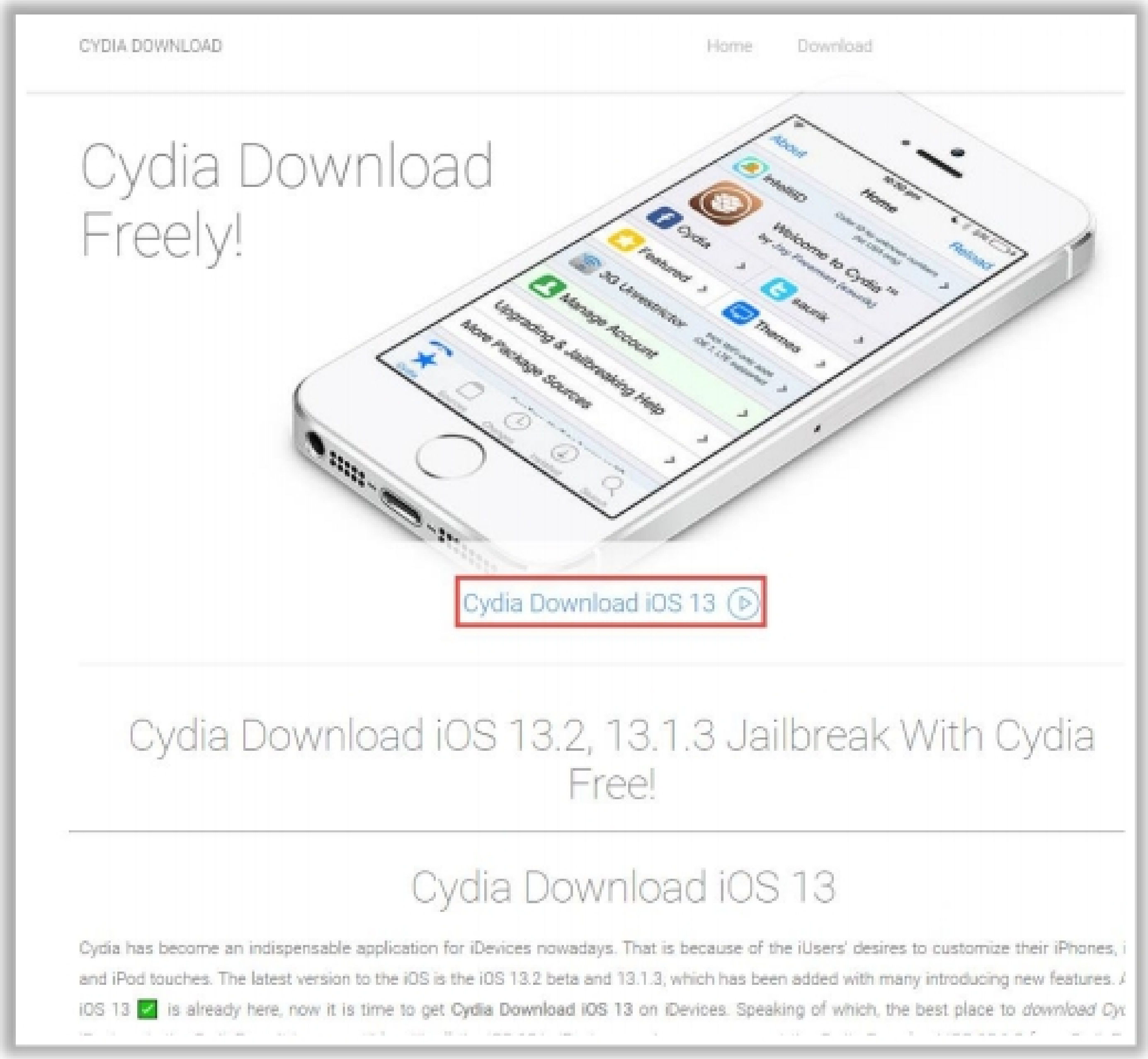


Figure 17.44: Enter URL

- Now, scroll down the website and click on the "**Cydia Download iOS 13**" button to go to the next page



Cydia Download Freely!

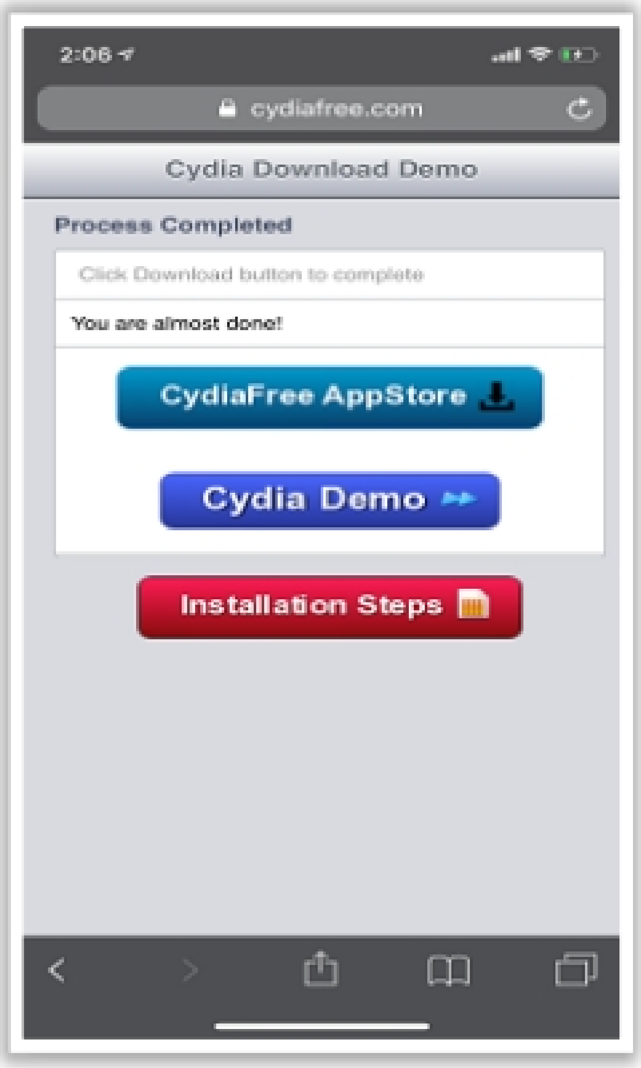
Cydia Download iOS 13.2, 13.1.3 Jailbreak With Cydia Free!

Cydia Download iOS 13

Cydia has become an indispensable application for iDevices nowadays. That is because of the iUsers' desires to customize their iPhones,
and iPod touches. The latest version to the iOS is the iOS 13.2 beta and 13.1.3, which has been added with many introducing new features.
iOS 13 ✓ is already here, now it is time to get **Cydia Download iOS 13** on iDevices. Speaking of which, the best place to *download*

Figure 17.45: Locate Up Arrow

- Click on the "Cydia Download iOS 13" button on that page to visit the download page
- On the Cydia download page, you will be given the necessary instructions to install Cydia iOS 13. Follow those instructions in the correct order to complete the installation

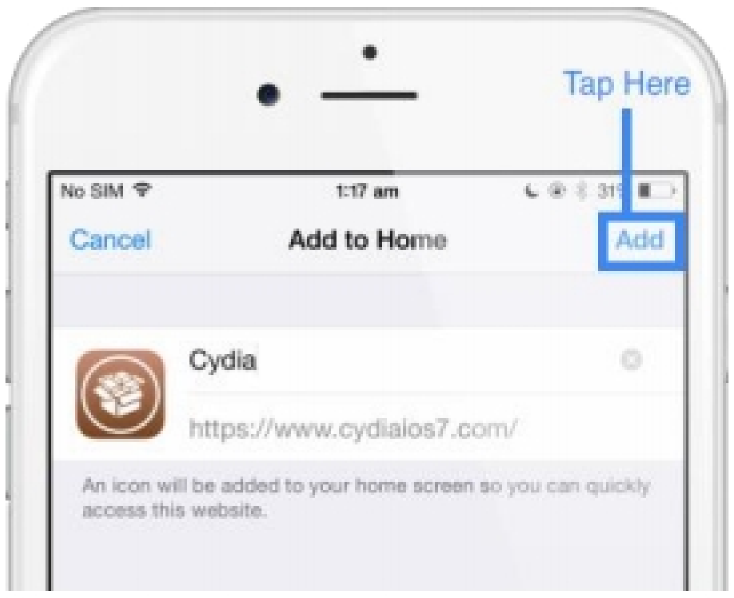

2:06
cydiafree.com
Cydia Download Demo
Process Completed
CydiaFree AppStore
Cydia Demo
Installation Steps

Click Download button to complete

You are almost done!

Figure 17.46: Cydia Download Page

- Now, type **Cydia** into the box for naming the app icon. Tap the **Add** button and close the Safari browser.


Tap Here
No SIM
1:17 am
Cancel
Add to Home
Add

Cydia

https://www.cydiaios7.com/

An icon will be added to your home screen so you can quickly access this website.

Figure 17.47: Name the App

- Locate the **Cydia** icon on your home screen

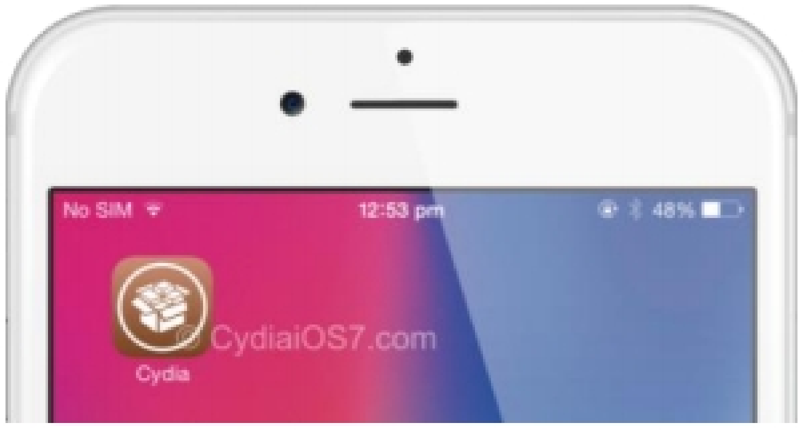

No SIM
12:53 pm
48%
CydiaiOS7.com
Cydia

Figure 17.48: Cydia icon on the home screen

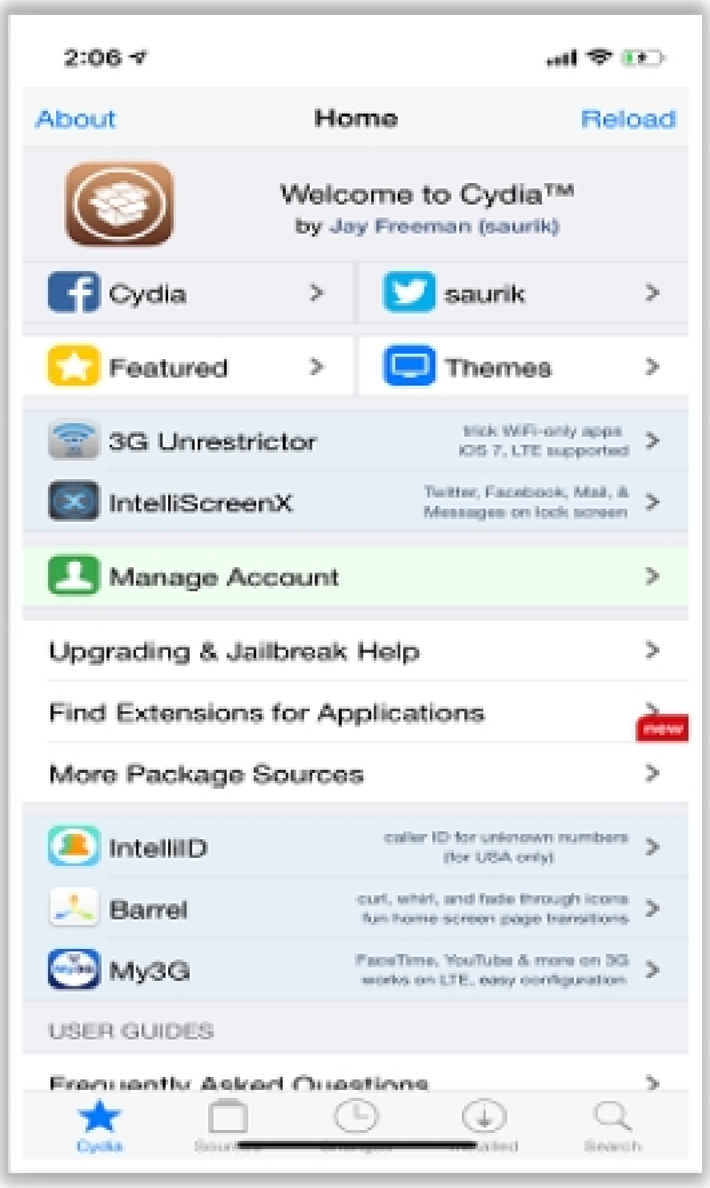

2:06
About
Home
Reload
Welcome to Cydia™
by Jay Freeman (saurik)
Cydia
saurik
Featured
Themes
3G Unrestrictor
trick WiFi-only apps
iOS 7, LTE supported
IntelliScreenX
Twitter, Facebook, Mail, &
Messages on lock screen
Manage Account
Upgrading & Jailbreak Help
Find Extensions for Applications
new
More Package Sources
IntelliID
caller ID for unknown numbers
(for USA only)
Barrel
curl, whirl, and fade through icons
fun home screen page transitions
My3G
FaceTime, YouTube & more on 3G
works on LTE, easy configuration
USER GUIDES
Frequently Asked Questions
Cydia
Sources
Changes
Installed
Search

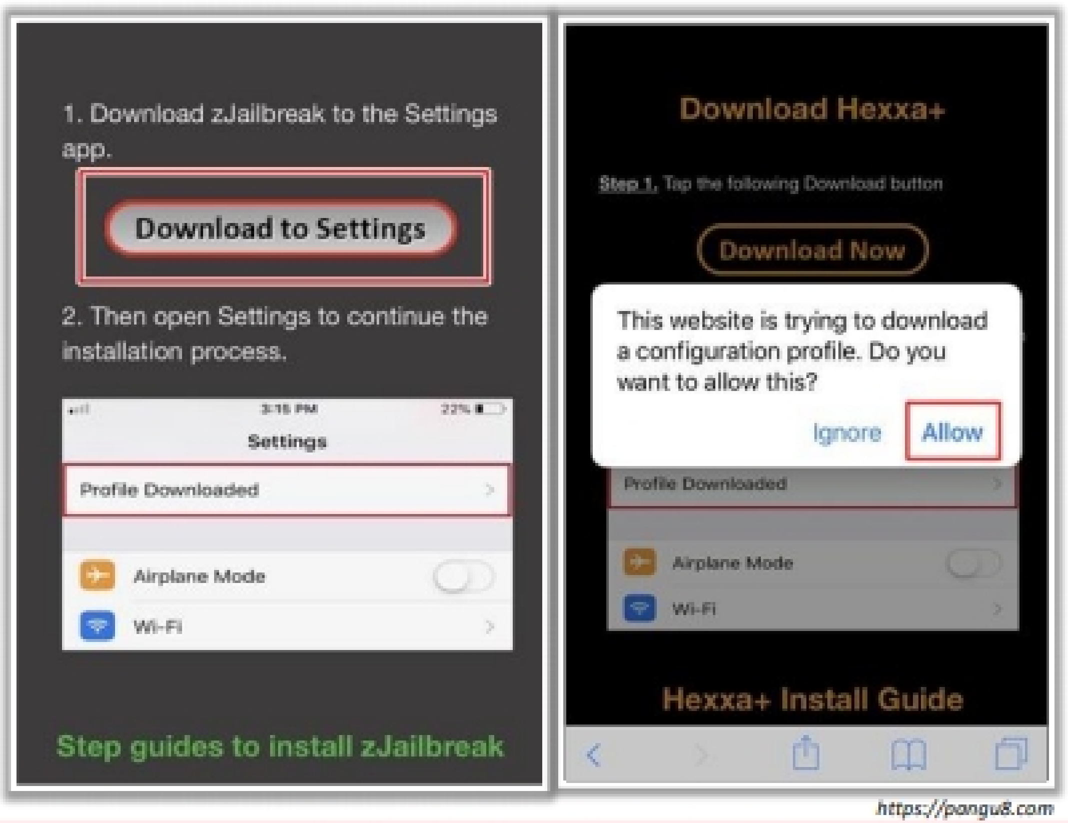
Figure 17.49: Cydia home screen

Jailbreaking of iOS 13.2 Using Hexxa Plus

- Hexxa Plus is a **Jailbreak Repo Extractor** for iOS 13.2, that allows you to install themes, tweaks, and apps
- It is compatible with iOS 13 and higher versions upto iOS 13.2.3, including iOS 13.3 beta

Steps to Install Hexxa Plus

1. Download **zJailbreak** app
2. Open the **zJailbreak** app. Go to **Hexxa Plus** app available under "jailbreak" by clicking on it
3. Click on "**Allow**" and then "**Download**"
4. Now, go to your device's "**Settings**" and click on "**Profile Download**"
5. Enter the **device passcode** to complete the Hexxa Plus installation process
6. Once you complete the installation process, the Hexxa Plus icon will appear on your **home screen**



1. Download zJailbreak to the Settings app.

2. Then open Settings to continue the installation process.

Step guides to install zJailbreak

Download Hexxa+

Step 1. Tap the following Download button

This website is trying to download a configuration profile. Do you want to allow this?

Hexxa+ Install Guide

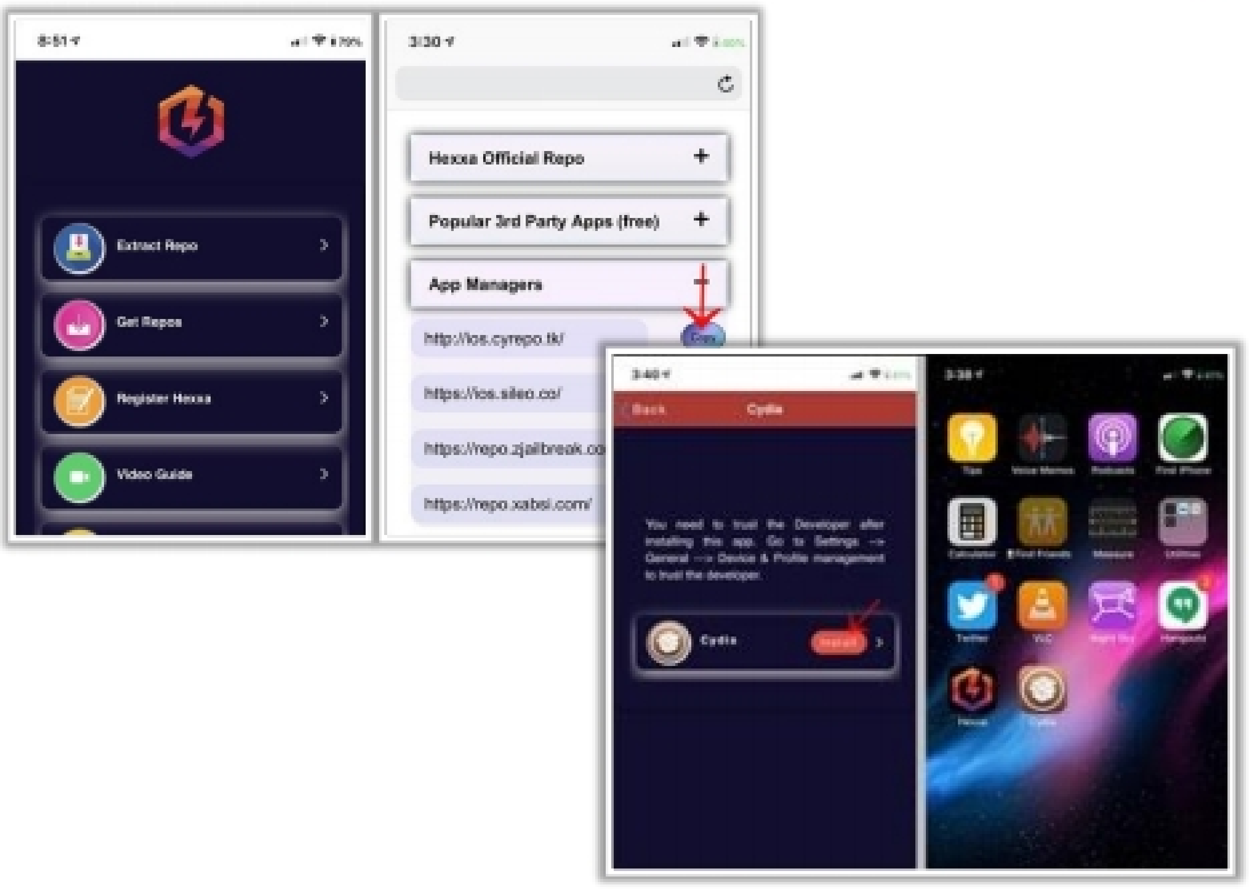
Download to Settings
Settings
Profile Downloaded
Airplane Mode
Wi-Fi
Download Now
Ignore
Allow

https://pangu8.com

Jailbreaking of iOS 13.2 Using Hexxa Plus (Cont'd)

Steps to Install Cydia Jailbreak Apps

1. Open the **Hexxa Plus** app → **Get Repos** → go to **App Managers**
2. Click the **Copy** button on ***http://ios.cyrepo.tk/*** Cydia repo
3. Go to **Extract Repo** and select it. Paste the copied Cydia repo
4. Extract the Cydia repo by clicking the **OK** button. Now, tap the **Install** button to install Cydia
5. Finally, you can see the Cydia app icon on your **home screen**



You need to trust the Developer after installing this app. Go to Settings --> General --> Device & Profile management to trust the developer.

Extract Repo
Get Repos
Register Hexxa
Video Guide
Hexxa Official Repo
Popular 3rd Party Apps (free)
App Managers
http://ios.cyrepo.tk/
Copy
https://ios.sileo.co/
https://repo.zjailbreak.co
https://repo.xabsi.com/
Back
Cydia
Install

Jailbreaking iOS 13.2 Using Hexxa Plus

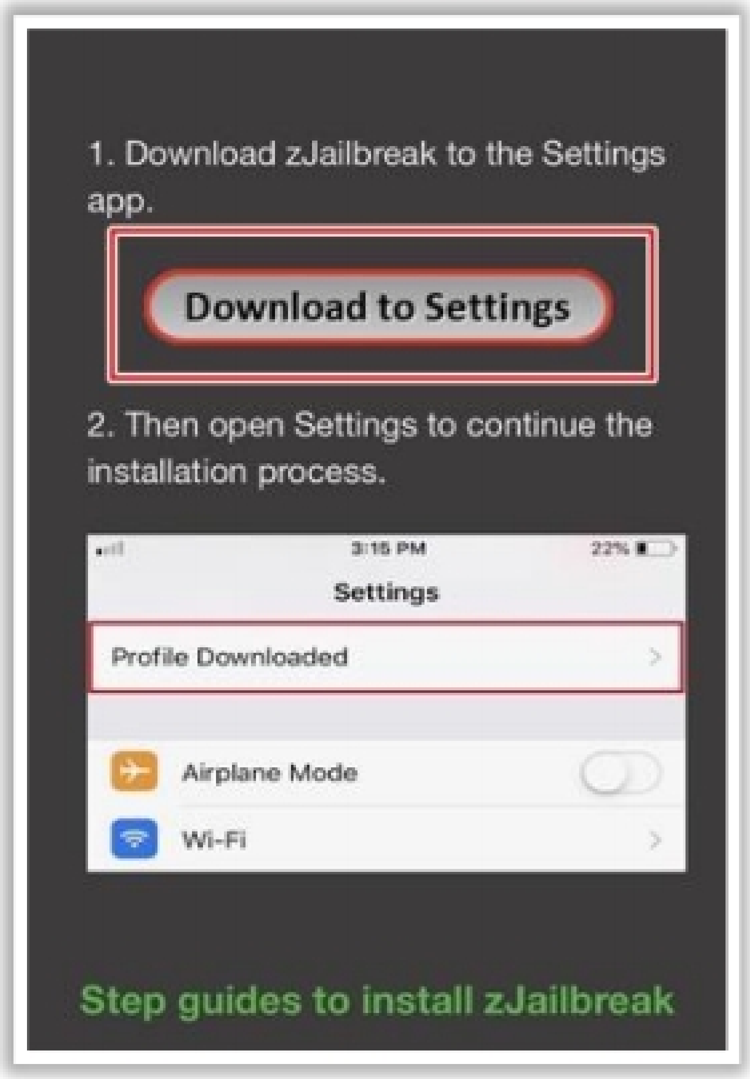
Source: *https://pangu8.com*

Hexxa Plus is a Jailbreak Repo Extractor for iOS 13.2, which allows you to install themes, tweaks, and apps. It is compatible with iOS 13 and higher versions up to iOS 13.2.3 including iOS 13.3 beta. This is the most popular method to install jailbreak apps without untethered or semi-untethered jailbreak. You can install iOS 13.2 jailbreak apps by extracting repos. There are many jailbreak repositories under Hexxa plus with thousands of jailbreak tweaks, themes, games, and so on.

Hexxa Plus allows you to install Cydia and popular Jailbreak apps to your latest iOS versions via the developer code extraction method. You must install a third-party app manager such as zJailbreak to install Hexxa Plus.

Steps to Install Hexxa Plus

- **Step 1**—Download the **zJailbreak** app. Provide your device passcode for this step.



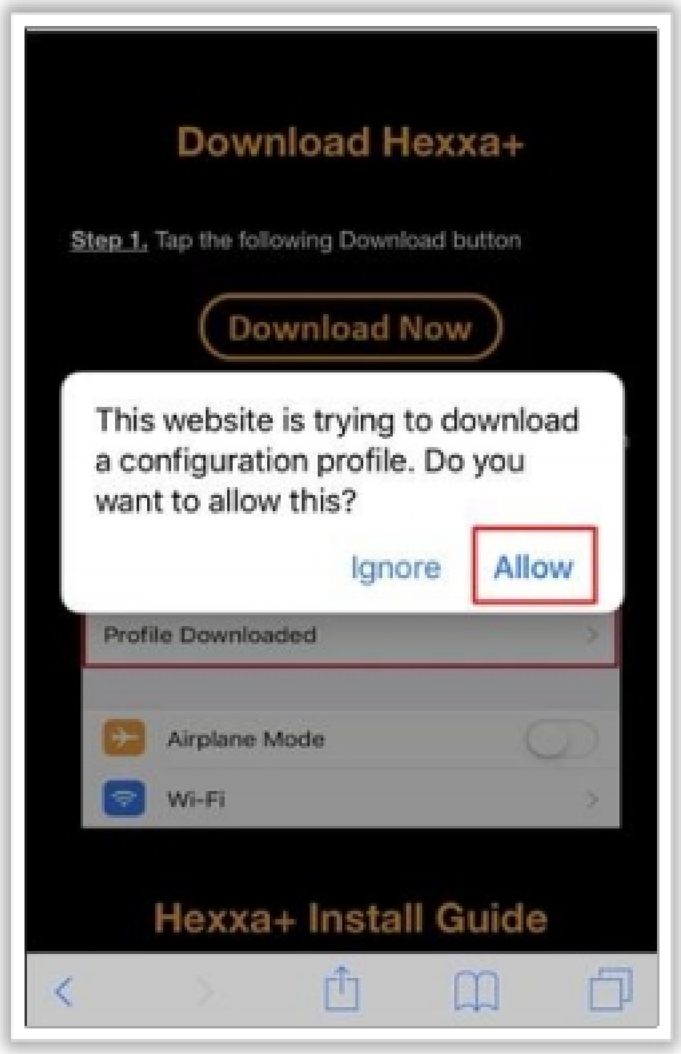
1. Download zJailbreak to the Settings app.

2. Then open Settings to continue the installation process.

Step guides to install zJailbreak

Figure 17.50: zJailbreak Download page

- **Step 2**—Open the **zJailbreak** app. Go to the **Hexxa Plus** app available under jailbreak by clicking on it
- **Step 3**— Click on “**Allow**”, then “**Download**.” **Hexxa ++** profile will be downloaded to the iOS 13.2 device settings.


Download Hexxa+
Step 1. Tap the following Download button
Download Now
Ignore
Allow
Profile Downloaded
Airplane Mode
Wi-Fi
Hexxa+ Install Guide

This website is trying to download a configuration profile. Do you want to allow this?

Figure 17.51: Hexxa Plus Download page

- **Step 4**— Now, go to your device “**Settings**”; then click on “**Profile Download**”.
- **Step 5** — Enter the device passcode to complete the **Hexxa Plus** installation process
- **Step 6**—Once you complete the installation process, the **Hexxa Plus** icon will appear on your home screen

Steps to Install Cydia Jailbreak Apps

- **Step 01**—Open the **Hexxa Plus** app → **Get Repos** → go to **App Managers**

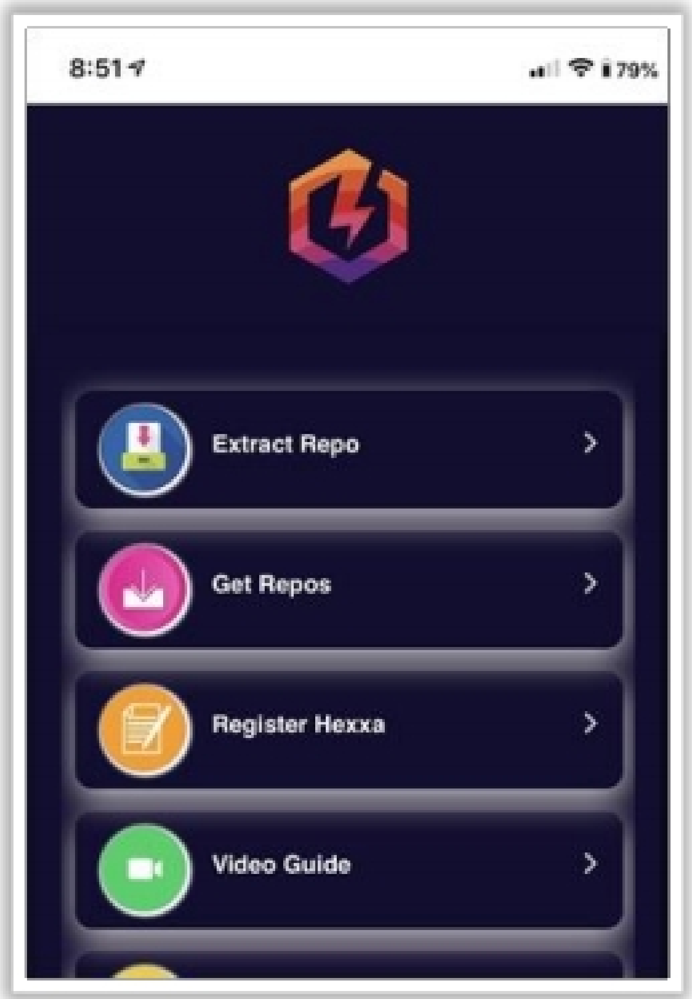

8:51
79%
Extract Repo
Get Repos
Register Hexxa
Video Guide

Figure 17.52: Screenshot of Hexxa Plus app

- **Step 02**— Click the **Copy** button on ***http://ios.cyrepo.tk/*** Cydia repo

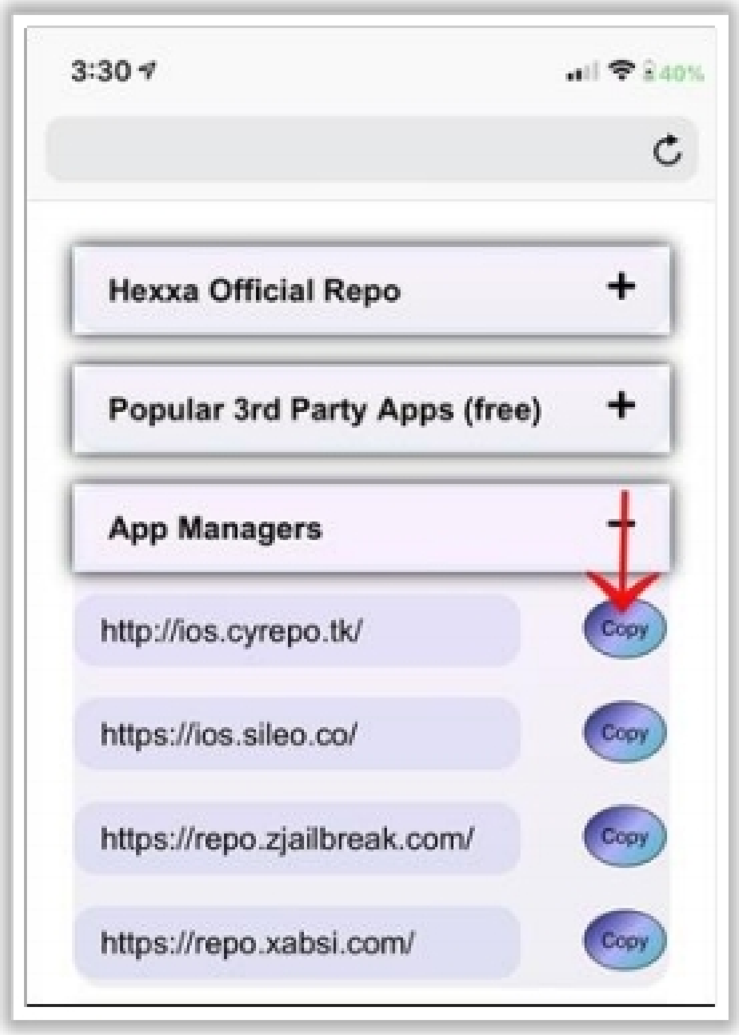
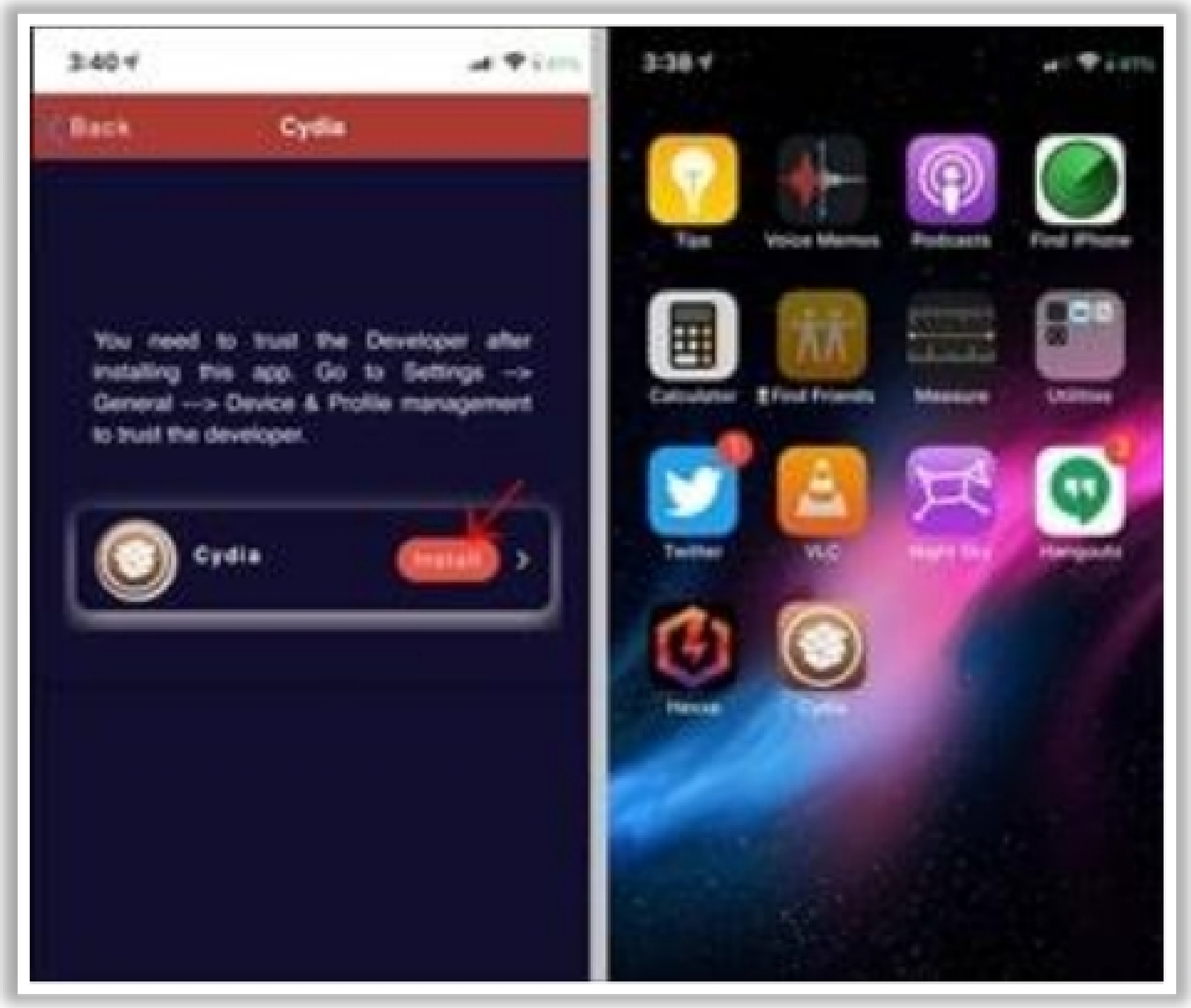

3:30
40%
Hexxa Official Repo +
Popular 3rd Party Apps (free) +
App Managers +
http://ios.cyrepo.tk/
Copy
https://ios.sileo.co/
Copy
https://repo.zjailbreak.com/
Copy
https://repo.xabsi.com/
Copy

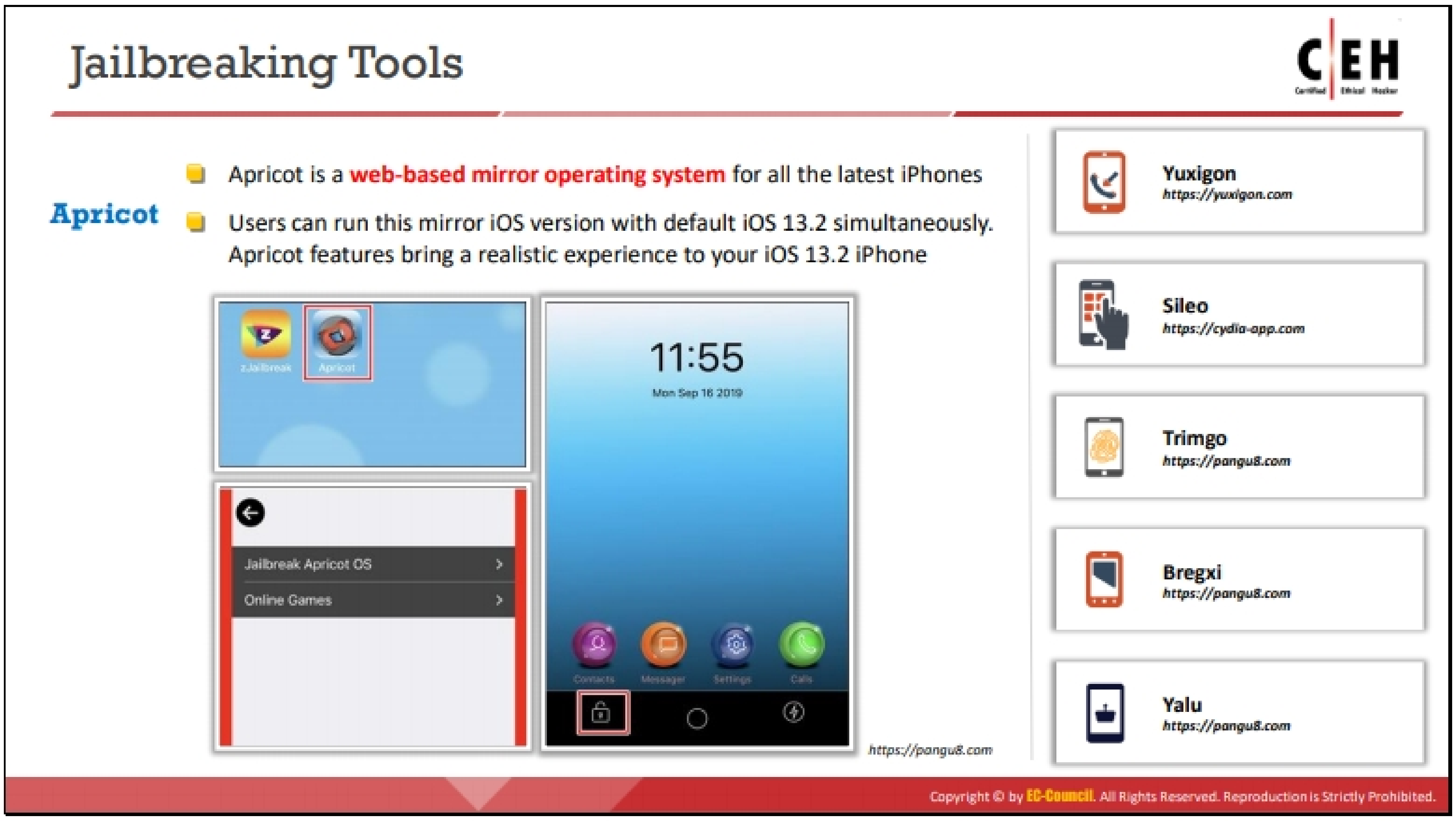
Figure 17.53: Hexxa Plus App Managers

- **Step 03**— Go to **Extract Repo** and choose it. Paste the copied Cydia repo
- **Step 04**— Extract the Cydia repo by clicking the **OK** button. Now, tap the **Install** button to install Cydia
- **Step 05**—Finally, you can see the Cydia app icon on your home screen


You need to trust the Developer after installing this app. Go to Settings --> General --> Device & Profile management to trust the developer.

3:40
Back
Cydia
Cydia
Install
3:38
Tips
Voice Memos
Podcasts
Find iPhone
Calculator
Find Friends
Measure
Utilities
Twitter
VLC
Night Sky
Hangouts
Hexxa
Cydia

Figure 17.54: Cydia Jailbreak application icon on the home screen



Jailbreaking Tools

Apricot

- Apricot is a **web-based mirror operating system** for all the latest iPhones
- Users can run this mirror iOS version with default iOS 13.2 simultaneously. Apricot features bring a realistic experience to your iOS 13.2 iPhone

zJailbreak
Apricot
Jailbreak Apricot OS
Online Games
11:55
Mon Sep 16 2019
Contacts
Messager
Settings
Calls

https://pangu8.com

Yuxigon
https://yuxigon.com

Sileo
https://cydia-app.com

Trimgo
https://pangu8.com

Bregxi
https://pangu8.com

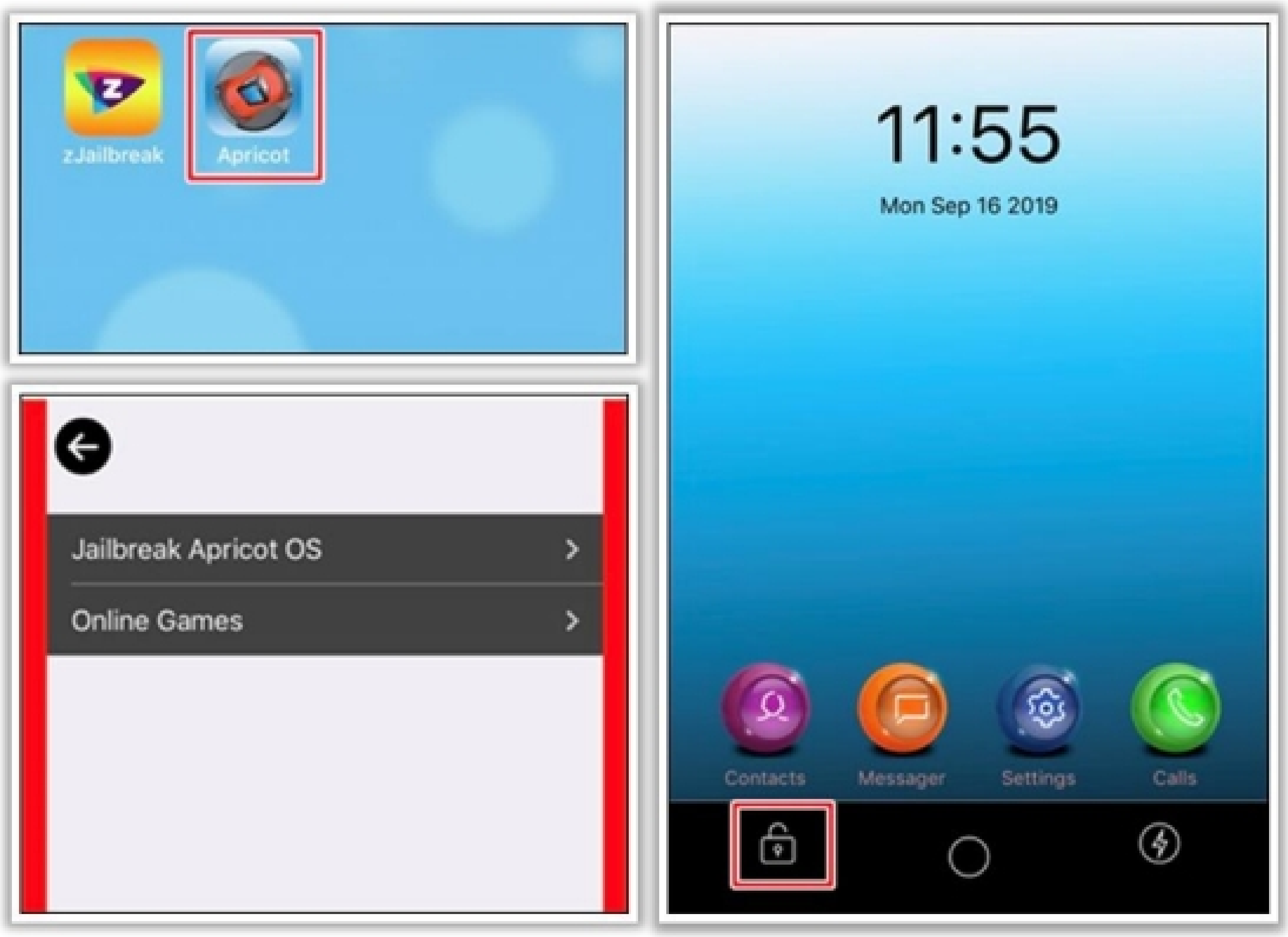
Yalu
https://pangu8.com

Jailbreaking Tools

- **Apricot**

  Source: *https://pangu8.com*

  Apricot is a web-based mirror OS for the latest iPhones. It supports iOS 13.2 devices. Users can run this mirror iOS version with the default iOS 13.2 simultaneously. Apricot features provide a realistic experience to your iOS 13.2 iPhone.



zJailbreak
Apricot
Jailbreak Apricot OS
Online Games
11:55
Mon Sep 16 2019
Contacts
Messager
Settings
Calls

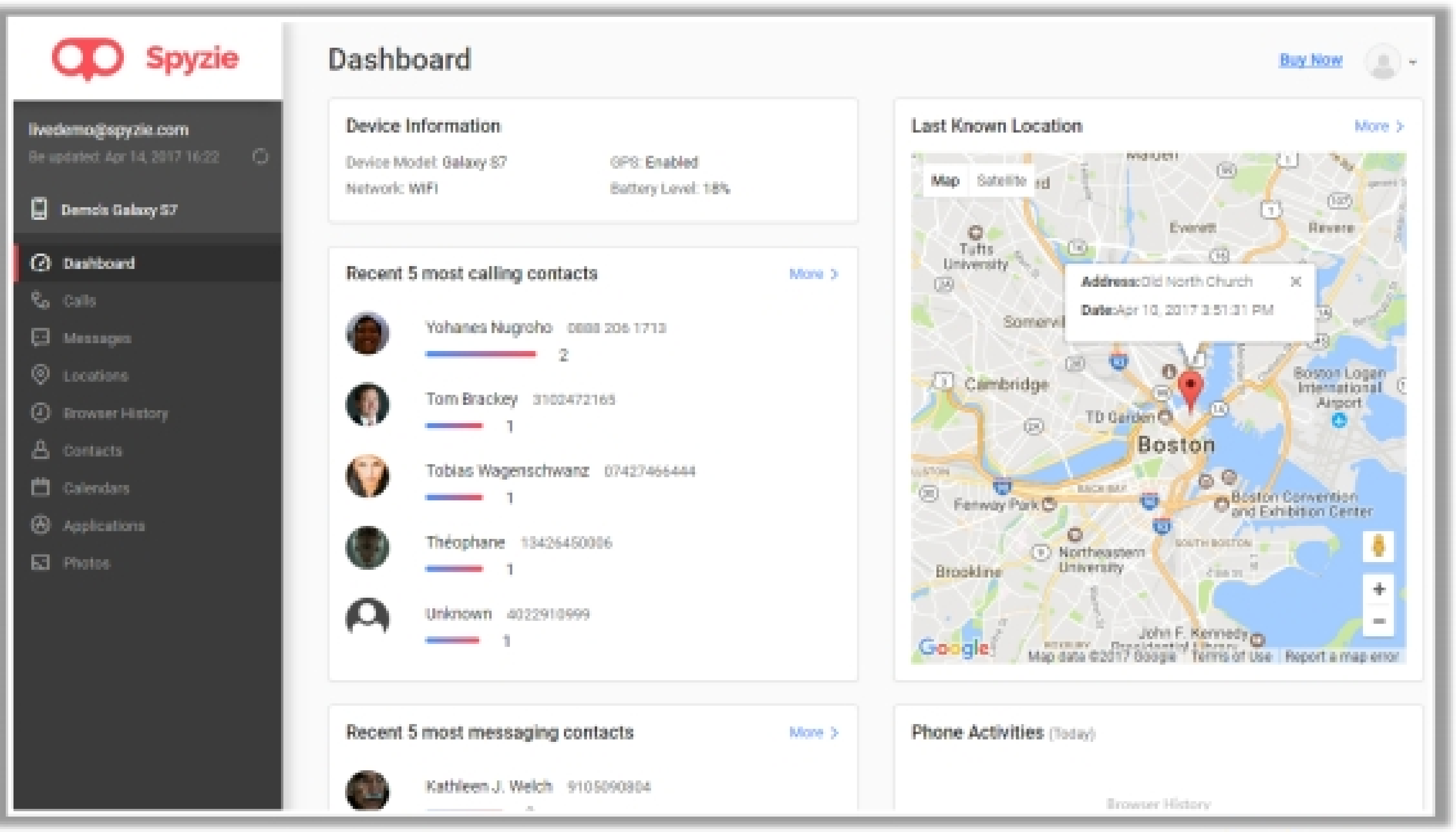
Figure 17.55: Screenshot of ApricotiOS

Some additional iOS Jailbreaking tools are as follows:

- Yuxigon (*https://yuxigon.com*)
- Sileo (*https://cydia-app.com*)
- Trimgo (*https://pangu8.com*)
- Bregxi (*https://pangu8.com*)
- Yalu (*https://pangu8.com*)

Hacking using Spyzie

- Spyzie allows attackers to **hack SMSs**, **call logs**, **app chats**, **GPS**, etc.
- This tool is compatible with all types of iOS devices, including iPhone, iPad, and iPod
- Attackers hack the target device remotely in an **invisible mode** without even jailbreaking the device



livedemo@spyzie.com
Be updated: Apr 14, 2017 16:22
Demo's Galaxy S7

Device Information
Device Model: Galaxy S7
Network: WiFi
GPS: Enabled
Battery Level: 18%

Recent 5 most calling contacts
Yohanes Nugroho 0888 206 1713 2
Tom Brackey 3102472165 1
Tobias Wagenschwanz 07427466444 1
Théophane 13426450006 1
Unknown 4022910999 1

Last Known Location
Address: Old North Church
Date: Apr 10, 2017 3:51:31 PM

Recent 5 most messaging contacts
Kathleen J. Welch 9105090804

Phone Activities (Today)

https://www.spyzie.com

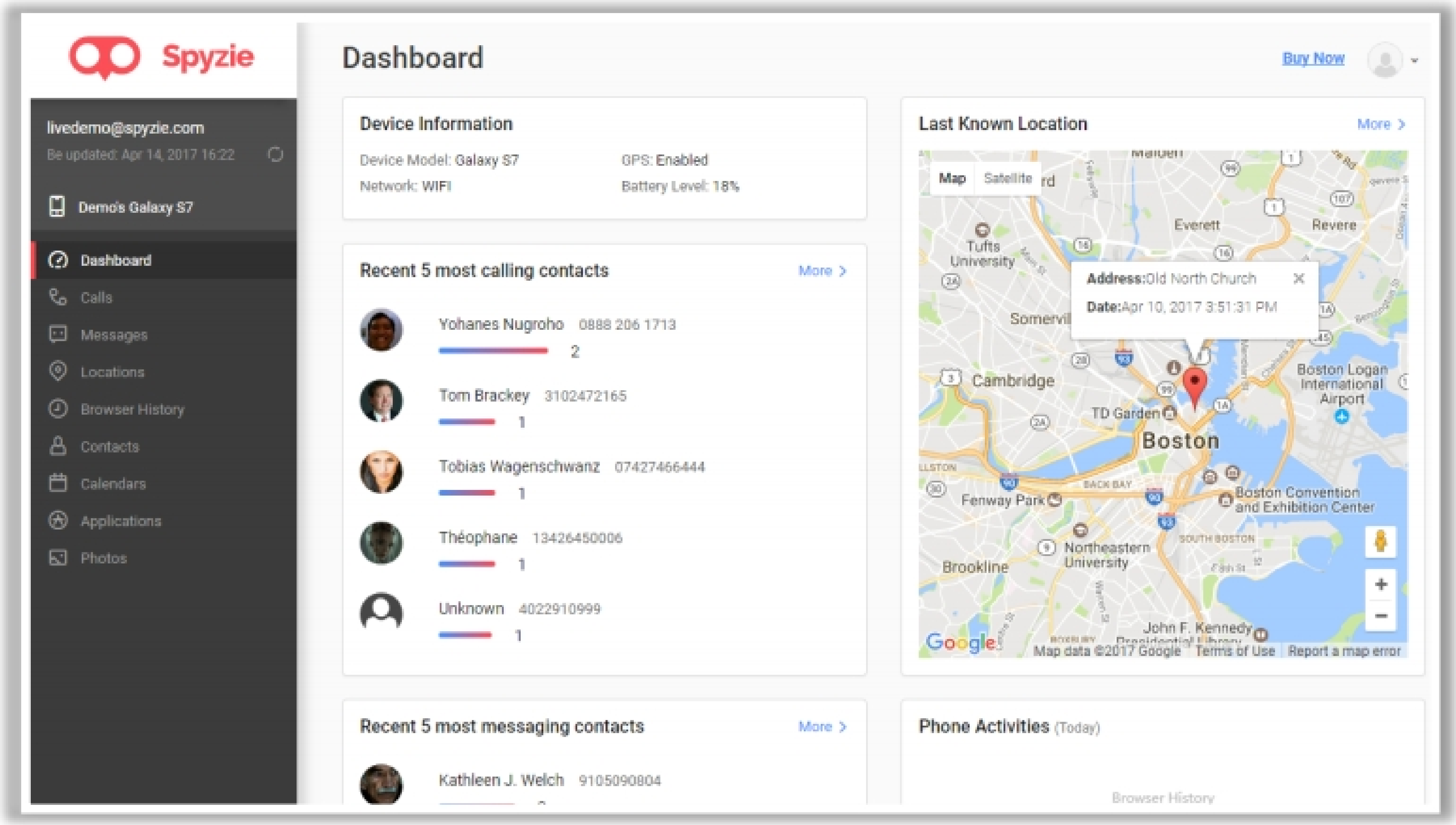
Hacking iOS Devices

Attackers use various methods to exploit iOS vulnerabilities. They use exploit chain tools that target various security vulnerabilities to penetrate different layers of iOS digital protection. They also install malicious software such as spyware and Trojans, to hack iOS devices.

Hacking using Spyzie

Source: *https://www.spyzie.com*

Attackers use various online tools such as Spyzie to hack the target iOS mobile devices. Spyzie allows attackers to hack SMS, call logs, app chats, GPS, etc. This tool is compatible with all types of iOS devices such as iPhone, iPad, and iPod. Attackers hack the target device remotely in an invisible mode without jailbreaking the device.


Spyzie
Dashboard
Buy Now
Dashboard
Calls
Messages
Locations
Browser History
Contacts
Calendars
Applications
Photos
More
Map
Satellite
Boston
Cambridge
Google

livedemo@spyzie.com
Be updated: Apr 14, 2017 16:22

Demo's Galaxy S7

Device Information

Device Model: Galaxy S7
GPS: Enabled
Network: WIFI
Battery Level: 18%

Recent 5 most calling contacts

Yohanes Nugroho 0888 206 1713 2
Tom Brackey 3102472165 1
Tobias Wagenschwanz 07427466444 1
Théophane 13426450006 1
Unknown 4022910999 1

Last Known Location

Address:Old North Church
Date:Apr 10, 2017 3:51:31 PM

Recent 5 most messaging contacts

Kathleen J. Welch 9105090804

Phone Activities (Today)

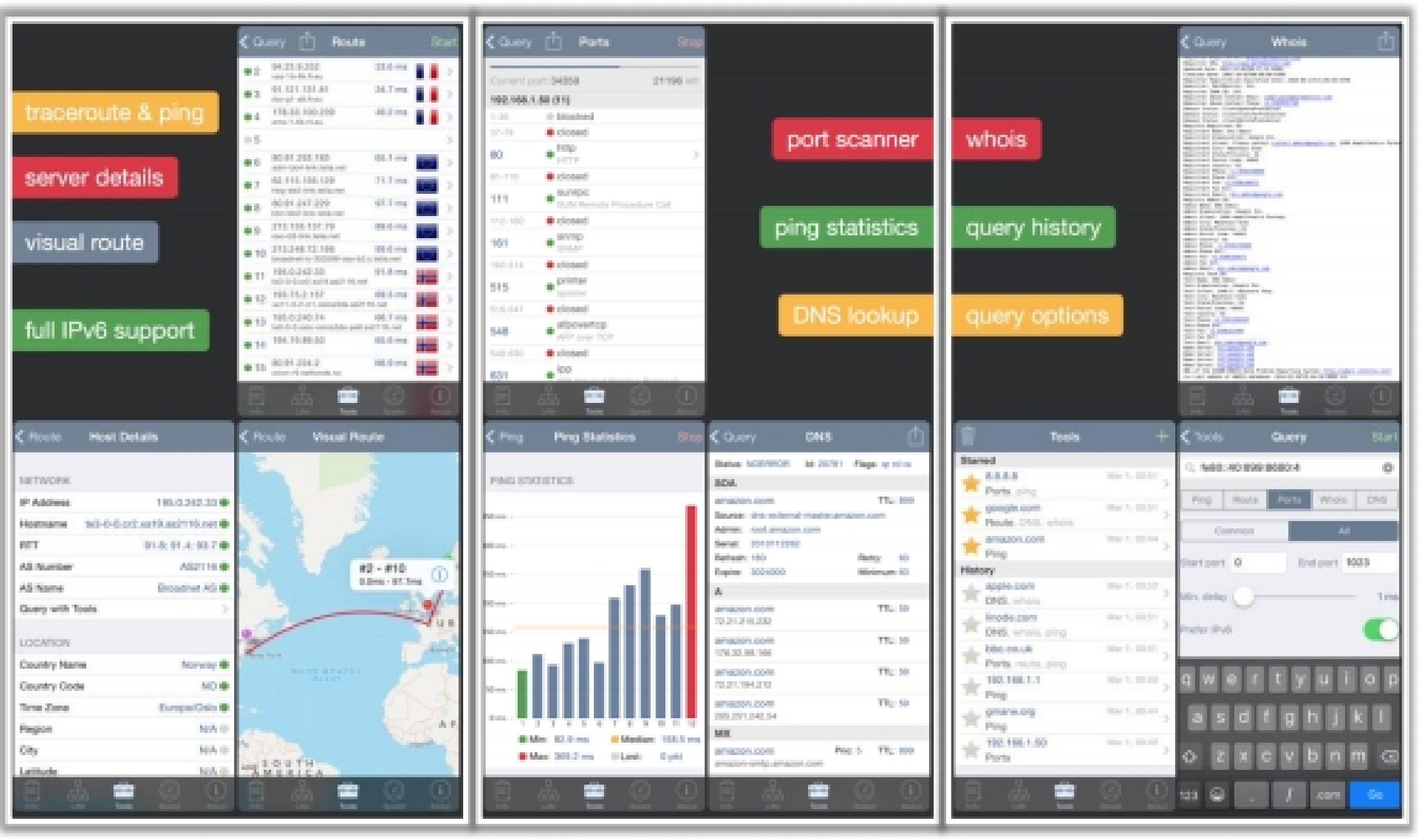
Browser History

Figure 17.56: Screenshot of Spyzie

Hacking Network using Network Analyzer Pro

- Network Analyzer Pro discovers all **LAN devices' addresses** and names
- Network Analyzer Pro allows attackers to gather information, such as devices connected to the network, their IP addresses, NetBIOS, mDNS (Bonjour), LLMNR, DNS name, etc.

https://apps.apple.com


traceroute & ping
server details
visual route
full IPv6 support
port scanner
ping statistics
DNS lookup
whois
query history
query options

Hacking Network using Network Analyzer Pro

Source: *https://apps.apple.com*

Network Analyzer Pro is a Wi-Fi device discovery tool that discovers all the LAN devices' addresses and names, together with the Bonjour/DLNA services they provide. It includes standard diagnostic tools such as ping, traceroute, port scanner, DNS lookup, and whois. Network Analyzer Pro allows attackers to gather information such as devices connected to the network, their IP addresses, NetBIOS, mDNS (Bonjour), LLMNR, and DNS name. It also helps in scanning the most common ports or user-specified port ranges and in the detection of closed, firewalled, and open ports.

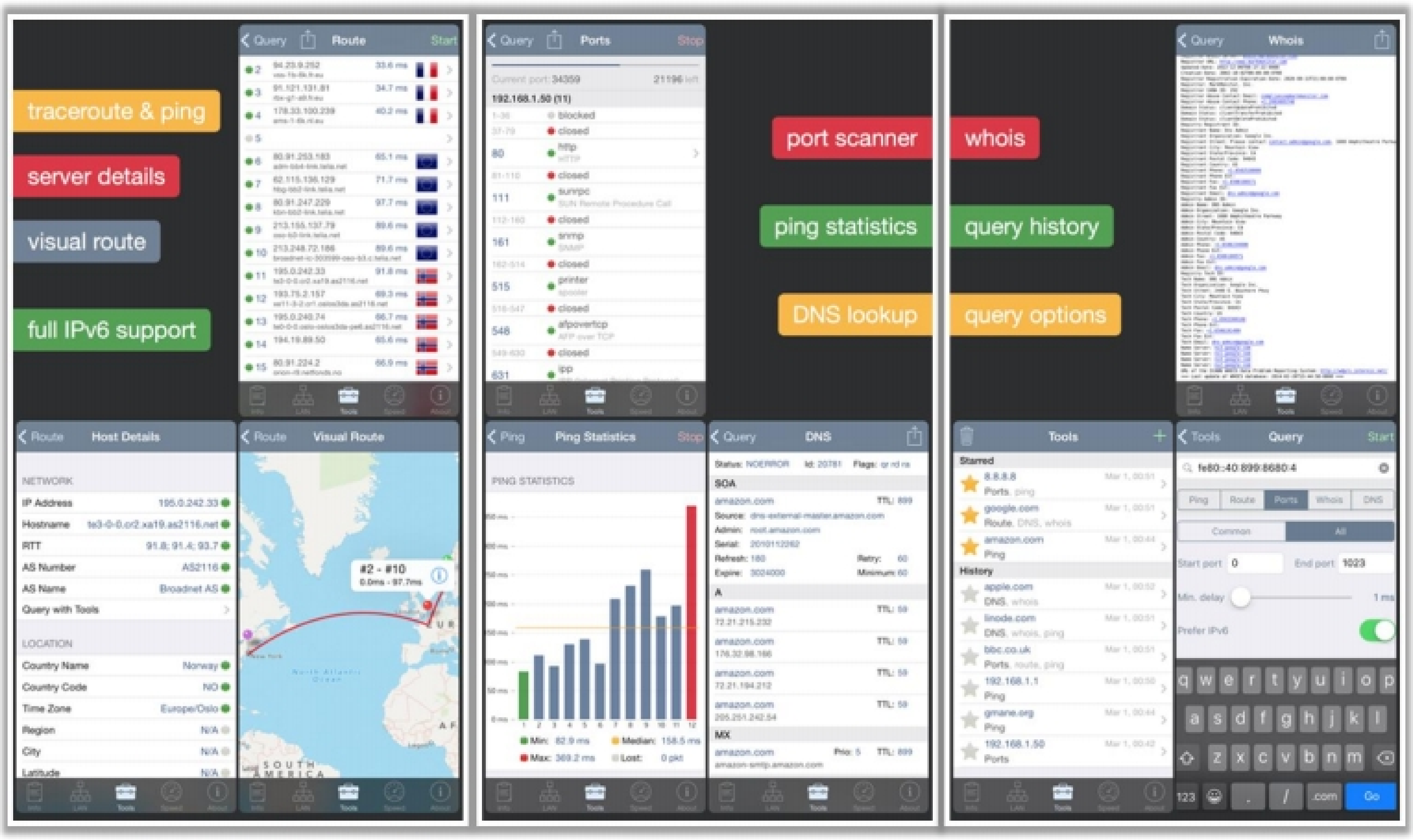
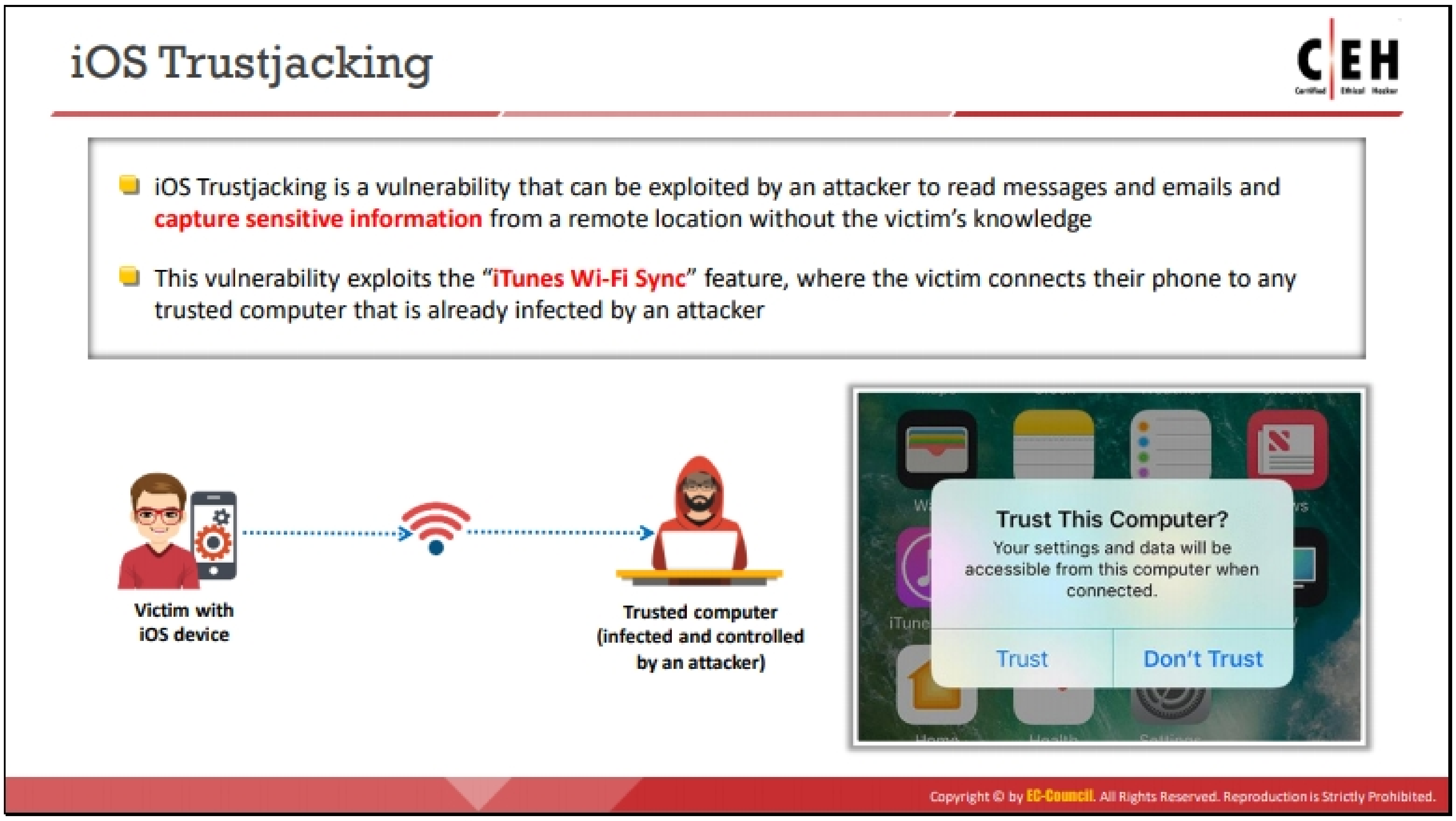

traceroute & ping
server details
visual route
full IPv6 support
port scanner
ping statistics
DNS lookup
whois
query history
query options

Figure 17.57: Screenshot of Network Analyzer Pro



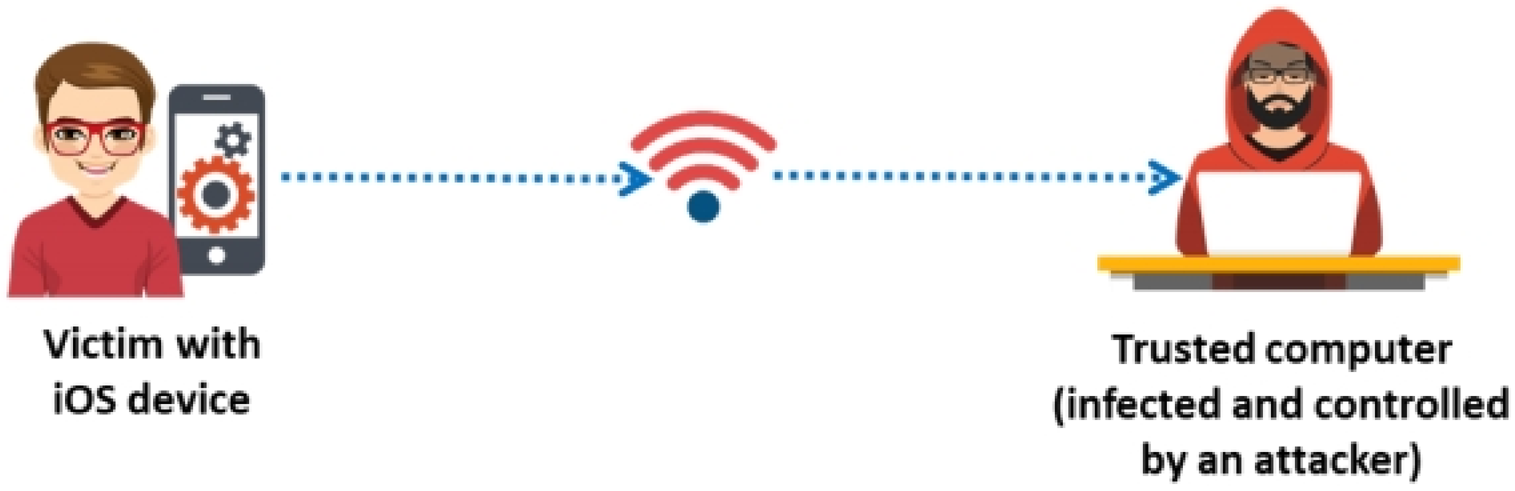
iOS Trustjacking

- iOS Trustjacking is a vulnerability that can be exploited by an attacker to read messages and emails and **capture sensitive information** from a remote location without the victim's knowledge
- This vulnerability exploits the "**iTunes Wi-Fi Sync**" feature, where the victim connects their phone to any trusted computer that is already infected by an attacker

Victim with iOS device
Trusted computer (infected and controlled by an attacker)
Trust This Computer?
Your settings and data will be accessible from this computer when connected.
Trust
Don't Trust

iOS Trustjacking

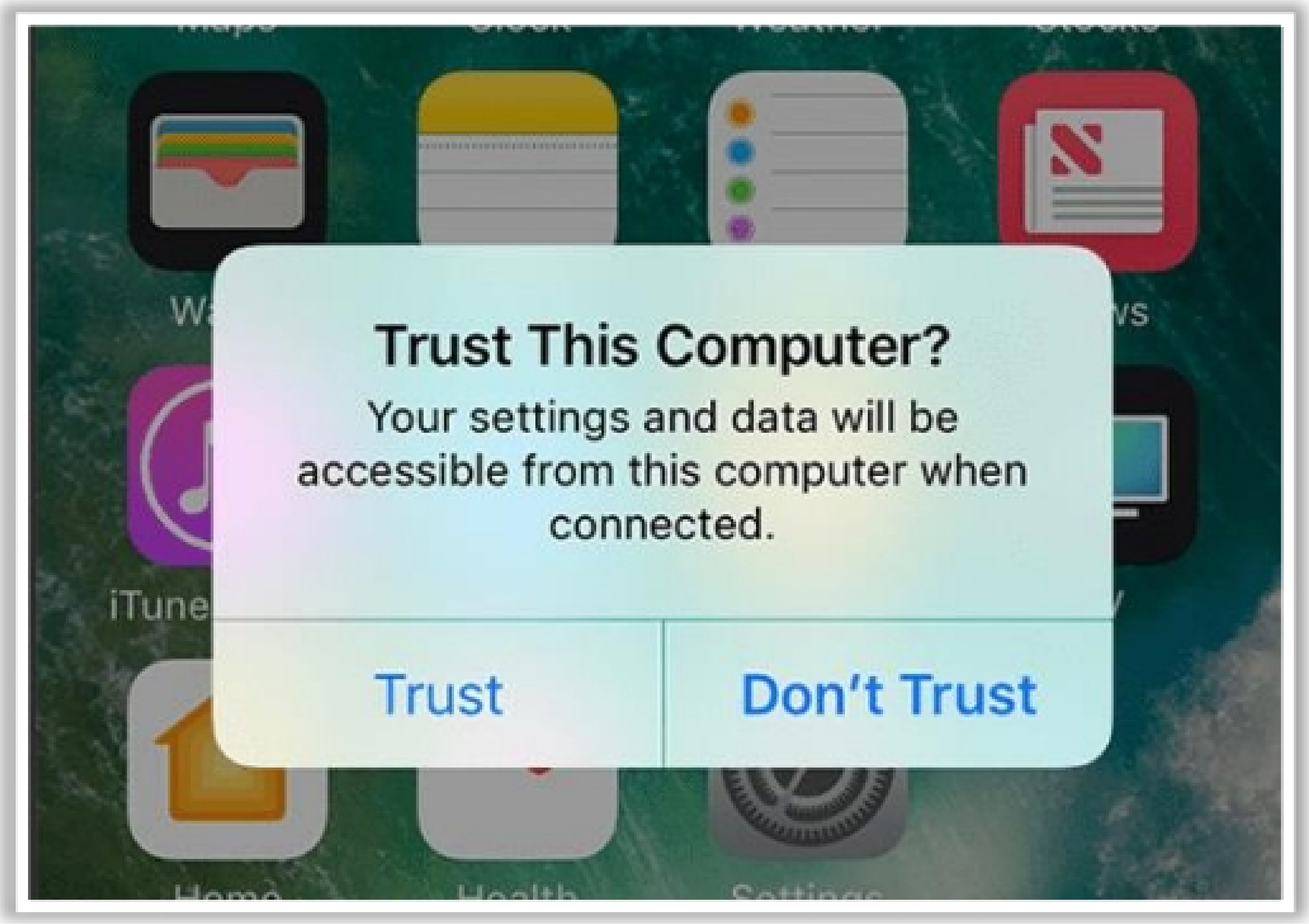
iOS Trustjacking is a vulnerability that can be exploited by an attacker to read messages and emails and capture sensitive information such as passwords and banking credentials from a remote location without a victim's knowledge. This vulnerability exploits the "iTunes Wi-Fi Sync" feature whereby a victim connects his/her phone to any trusted computer (could be of a friend or any trusted entity) that is already infected by the attacker.



Victim with iOS device
Trusted computer (infected and controlled by an attacker)

Figure 17.58: iOS Trustjacking attack

When an iOS device is trying to connect to a computer, the device displays a dialog box on the screen with the options "**Trust**" and "**Don't Trust**." Upon clicking **Trust**, it establishes a connection between the devices to share the information.



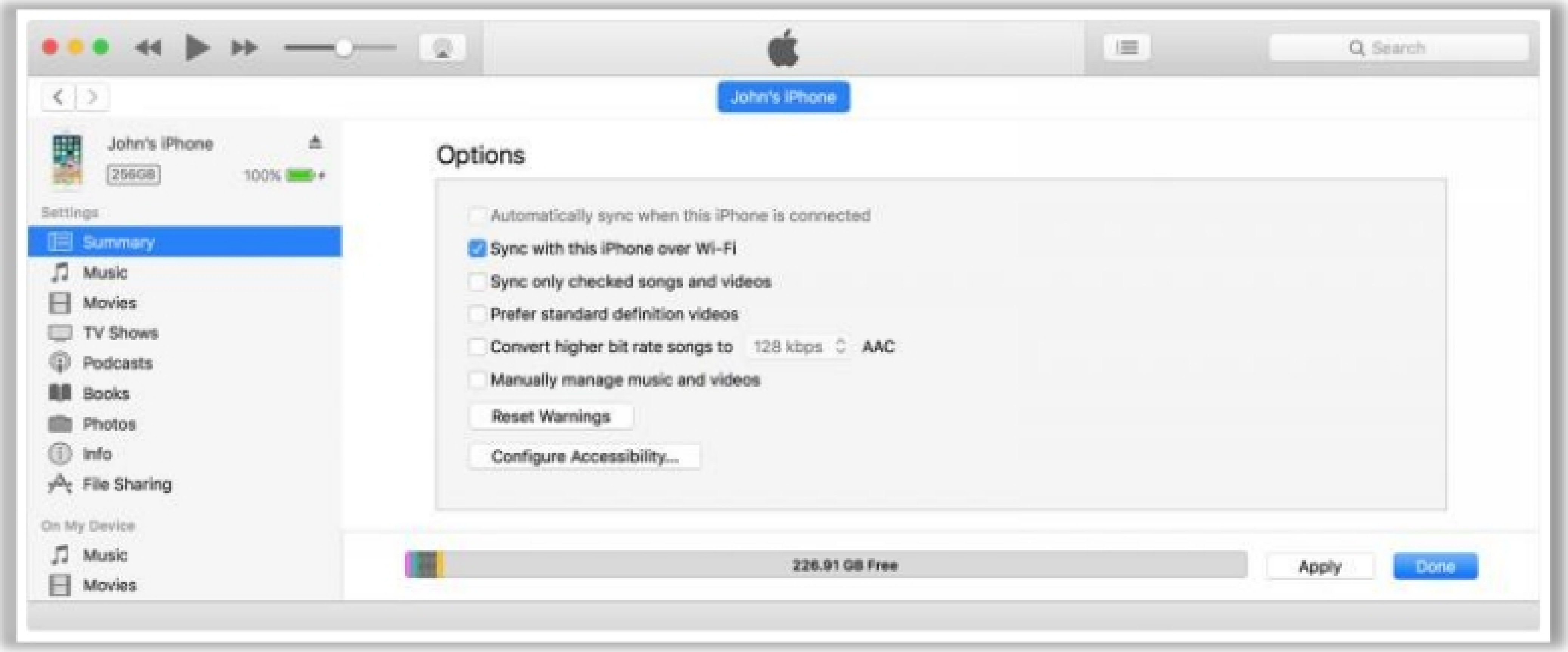
Trust This Computer?

Your settings and data will be accessible from this computer when connected.

Trust
Don't Trust

Figure 17.59: Screenshot of iOS demonstrating Trustjacking

After establishing the connection and enabling iTunes Wi-Fi sync on the computer, the device can continue its communication with that computer even after being physically disconnected.



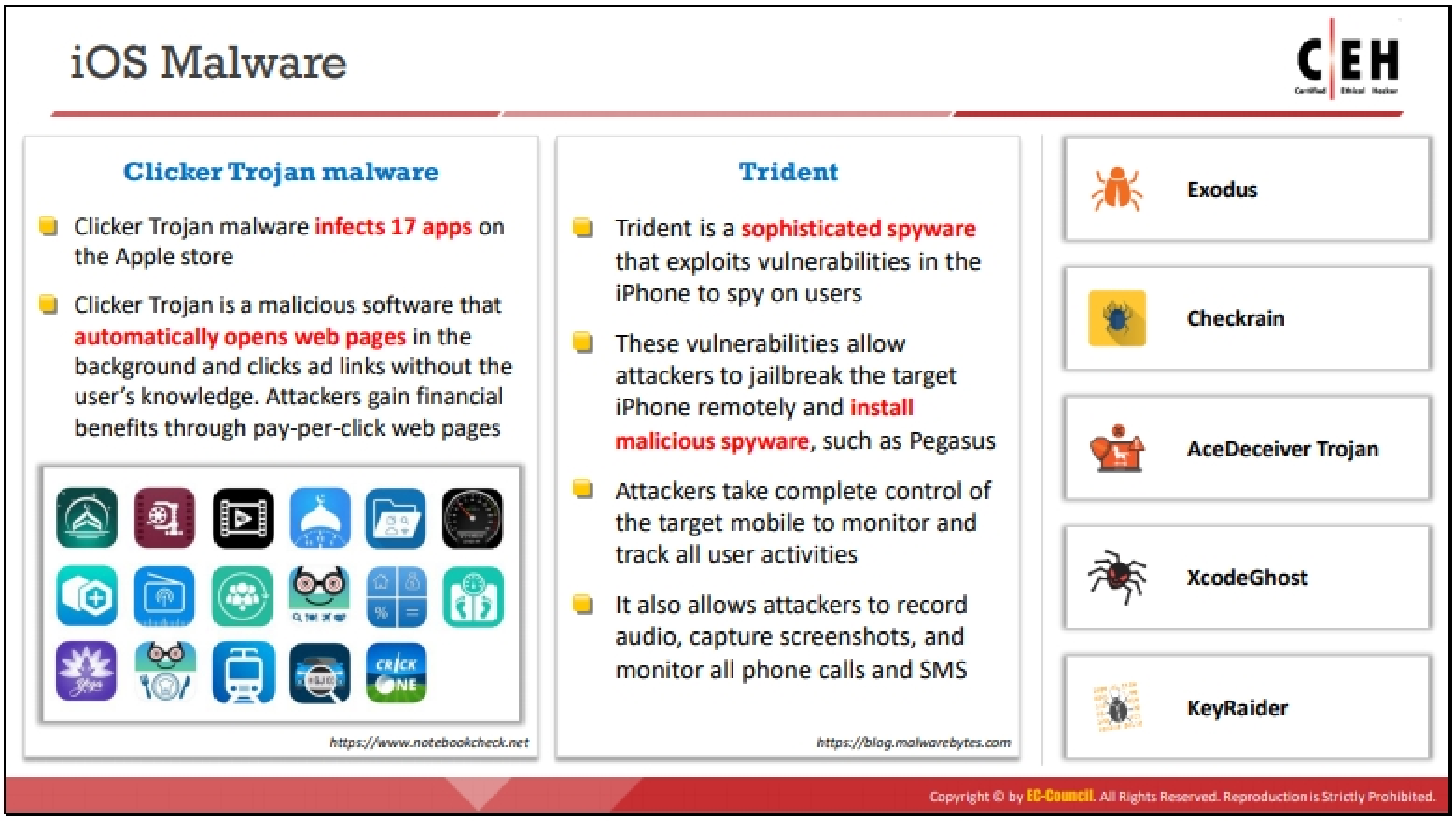
Options

☐ Automatically sync when this iPhone is connected
☑ Sync with this iPhone over Wi-Fi
☐ Sync only checked songs and videos
☐ Prefer standard definition videos
☐ Convert higher bit rate songs to 128 kbps AAC
☐ Manually manage music and videos

Search
John's iPhone
John's iPhone
256GB
100%
Settings
Summary
Music
Movies
TV Shows
Podcasts
Books
Photos
Info
File Sharing
On My Device
Music
Movies
Reset Warnings
Configure Accessibility...
226.91 GB Free
Apply
Done

Figure 17.60: Screenshot showing options for synchronizing iOS device and PC

Once the victim clicks on "**Trust**," the attacker gets access to the connected iOS device through the infected computer, which continues until the phone resets the connection settings. The data and screen operations of the compromised device can later be monitored from the desktop without the user's knowledge. The infected system can allow the attacker to read the user's activity even after the device is out of the communication zone. It can also enable the attacker to backup or restore data to read SMS history, deleted photos, and apps. The attacker can also replace original apps of the device with malicious apps from the previously connected PC.



iOS Malware

Clicker Trojan malware

- Clicker Trojan malware **infects 17 apps** on the Apple store
- Clicker Trojan is a malicious software that **automatically opens web pages** in the background and clicks ad links without the user's knowledge. Attackers gain financial benefits through pay-per-click web pages

https://www.notebookcheck.net

Trident

- Trident is a **sophisticated spyware** that exploits vulnerabilities in the iPhone to spy on users
- These vulnerabilities allow attackers to jailbreak the target iPhone remotely and **install malicious spyware**, such as Pegasus
- Attackers take complete control of the target mobile to monitor and track all user activities
- It also allows attackers to record audio, capture screenshots, and monitor all phone calls and SMS

https://blog.malwarebytes.com

- **Exodus**
- **Checkrain**
- **AceDeceiver Trojan**
- **XcodeGhost**
- **KeyRaider**

iOS Malware

- **Clicker Trojan malware**

  Source: *https://www.notebookcheck.net*

  Clicker Trojan malware has infected 17 apps on the Apple store. All these malicious apps use a clicker Trojan module to generate fake web traffic with the intention of generating ad revenue for specific websites. Clicker Trojan is a malicious software that automatically opens web pages in the background and clicks ad links without the user's knowledge. Attackers gain financial benefits through pay-per-click count of web pages.

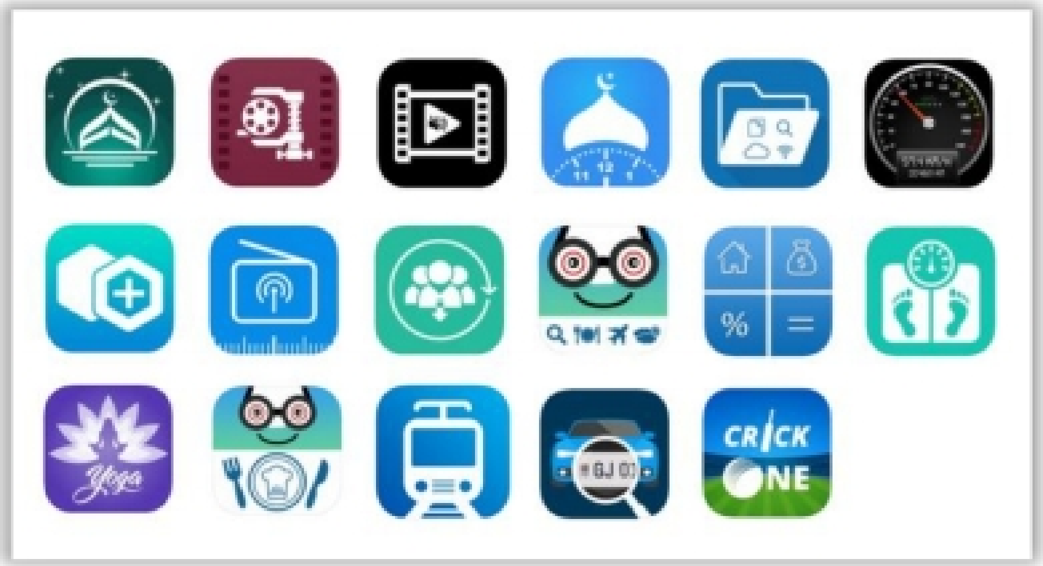


Figure 17.61: Clicker Trojan malware

- **Trident**

  Source: *https://blog.malwarebytes.com*

  Trident is a sophisticated spyware that exploits vulnerabilities in an iPhone to spy on users. These vulnerabilities allow attackers to jailbreak the target iPhone remotely and install malicious spyware such as Pegasus. Trident is capable of taking complete control of the target mobile device, and it allows attackers to monitor and track all the user activities. It also allows attackers to record audio, capture screenshots, and monitor all phone calls and SMS messages.

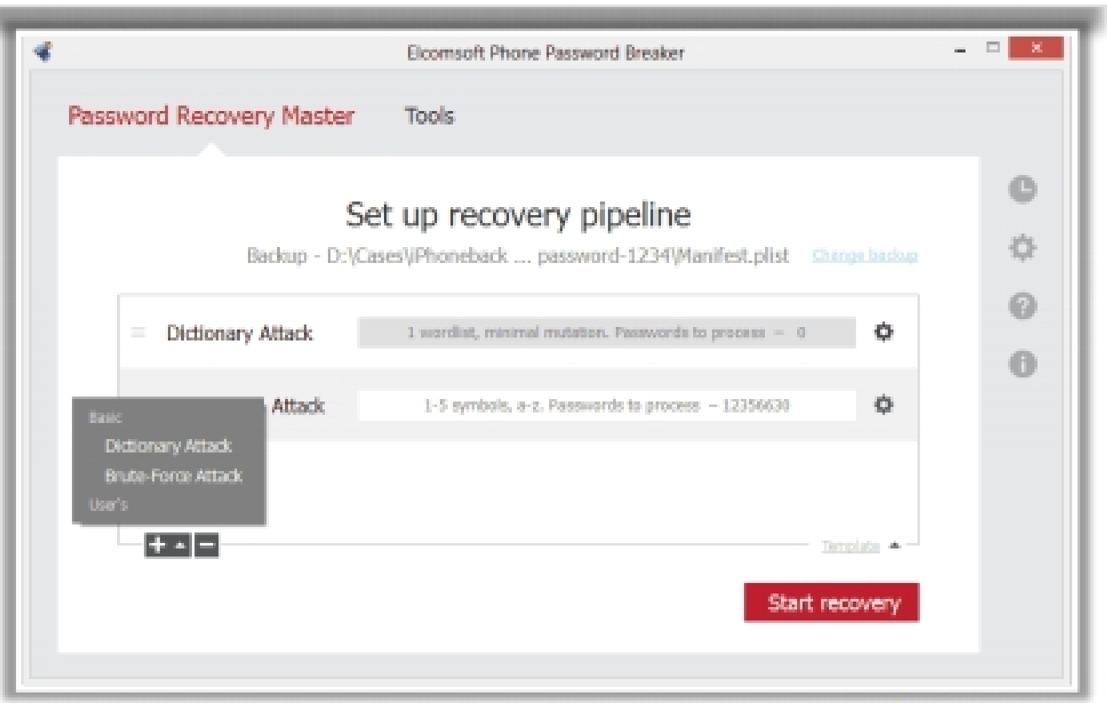
Some additional iOS malware are as follows:

- Exodus
- Checkrain
- AceDeceiver Trojan
- XcodeGhost
- KeyRaider

iOS Hacking Tools

Elcomsoft Phone Breaker

- Elcomsoft Phone Breaker allows attackers to perform logical and over-the-air acquisition of iOS devices, break into encrypted backups, and obtain and analyze backups, synchronized data, and passwords from Apple iCloud


Elcomsoft Phone Password Breaker
Password Recovery Master
Tools
Set up recovery pipeline
Backup - D:\Cases\iPhoneback ... password-1234\Manifest.plist
Dictionary Attack
1 wordlist, minimal mutation. Passwords to process ~ 0
Attack
1-5 symbols, a-z. Passwords to process ~ 12356630
Basic
Dictionary Attack
Brute-Force Attack
User's
Template
Start recovery

https://www.elcomsoft.com

Fing - Network Scanner
https://apps.apple.com

Network Analyzer Master Lite
https://apps.apple.com

Spyic
https://spyic.com

iWepPRO
https://apps.apple.com

Frida
https://www.frida.re

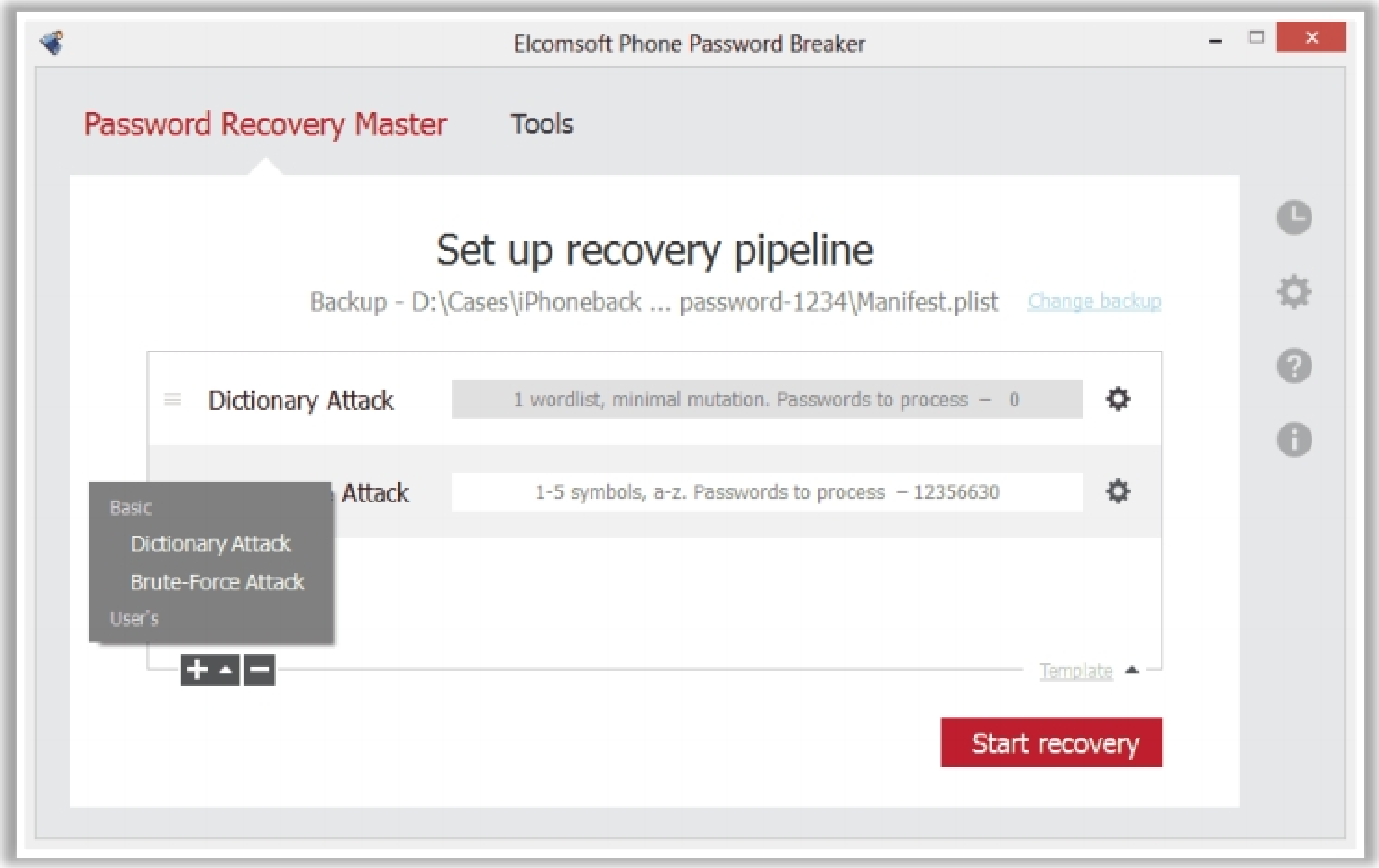
iOS Hacking Tools

Various tools used by attackers to hack target iOS mobile devices are discussed below:

- **Elcomsoft Phone Breaker**

  Source: *https://www.elcomsoft.com*

  Elcomsoft Phone Breaker allows attackers to perform logical and over-the-air acquisition of iOS devices, break into encrypted backups, and obtain and analyze backups, synchronized data, and passwords from Apple iCloud. It allows attackers to break passwords and decrypt iOS backups with GPU acceleration. Using this tool, attackers can decrypt iCloud Keychain and messages with media files and documents from iCloud.


Elcomsoft Phone Password Breaker
Password Recovery Master
Tools
Change backup
Basic
Dictionary Attack
Brute-Force Attack
User's
Template
Start recovery

Set up recovery pipeline

Backup - D:\Cases\iPhoneback ... password-1234\Manifest.plist

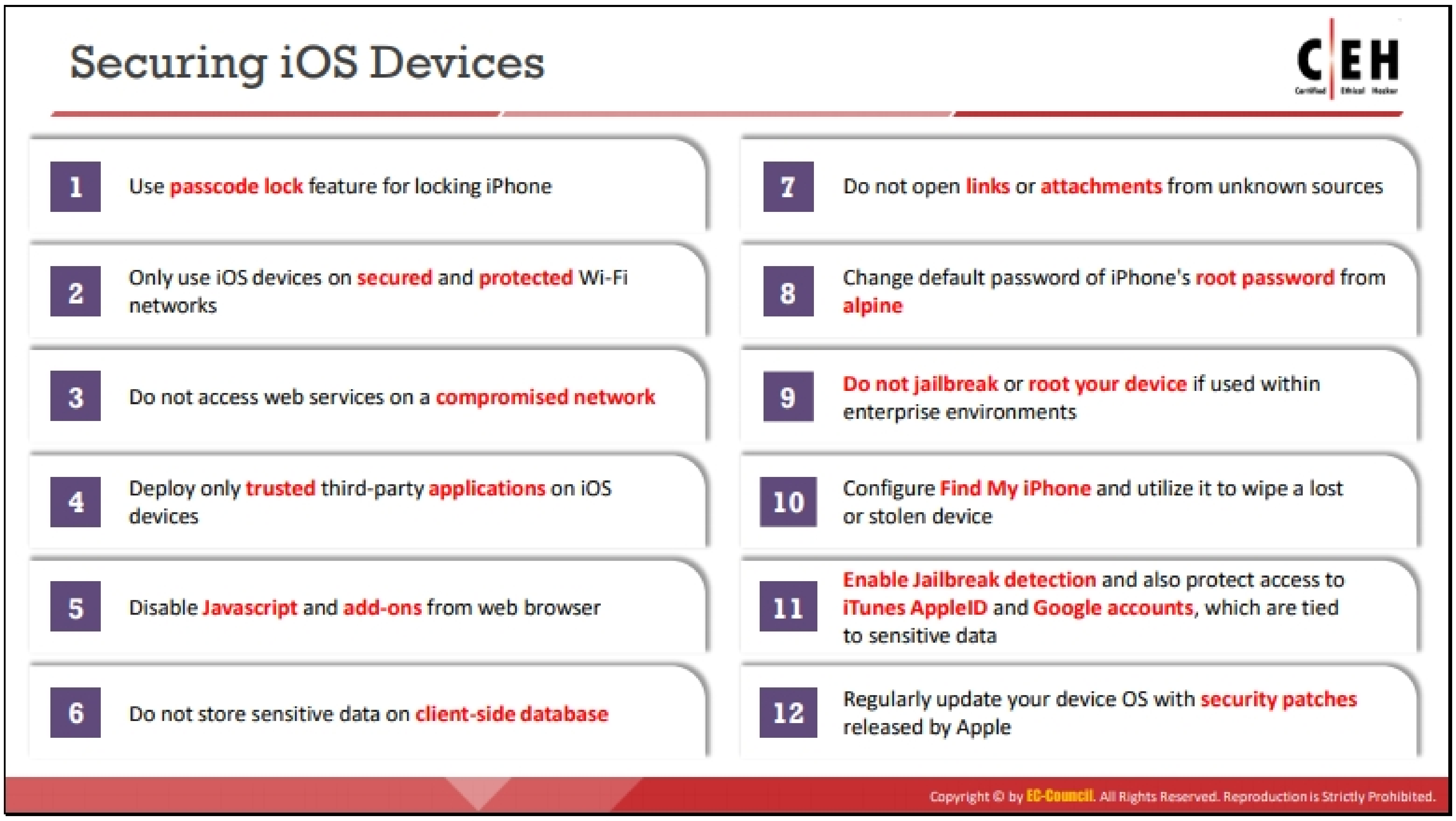
Dictionary Attack: 1 wordlist, minimal mutation. Passwords to process – 0

Attack: 1-5 symbols, a-z. Passwords to process – 12356630

Figure 17.62: Screenshot of Elcomsoft Phone Breaker

Some additional tools for hacking iOS devices are listed below:

- Fing - Network Scanner (*https://apps.apple.com*)
- Network Analyzer Master Lite (*https://apps.apple.com*)
- Spyic (*https://spyic.com*)
- iWepPRO (*https://apps.apple.com*)
- Frida (*https://www.frida.re*)



Securing iOS Devices

1. Use **passcode lock** feature for locking iPhone
2. Only use iOS devices on **secured** and **protected** Wi-Fi networks
3. Do not access web services on a **compromised network**
4. Deploy only **trusted** third-party **applications** on iOS devices
5. Disable **Javascript** and **add-ons** from web browser
6. Do not store sensitive data on **client-side database**
7. Do not open **links** or **attachments** from unknown sources
8. Change default password of iPhone's **root password** from **alpine**
9. **Do not jailbreak** or **root your device** if used within enterprise environments
10. Configure **Find My iPhone** and utilize it to wipe a lost or stolen device
11. **Enable Jailbreak detection** and also protect access to **iTunes AppleID** and **Google accounts**, which are tied to sensitive data
12. Regularly update your device OS with **security patches** released by Apple

Securing iOS Devices

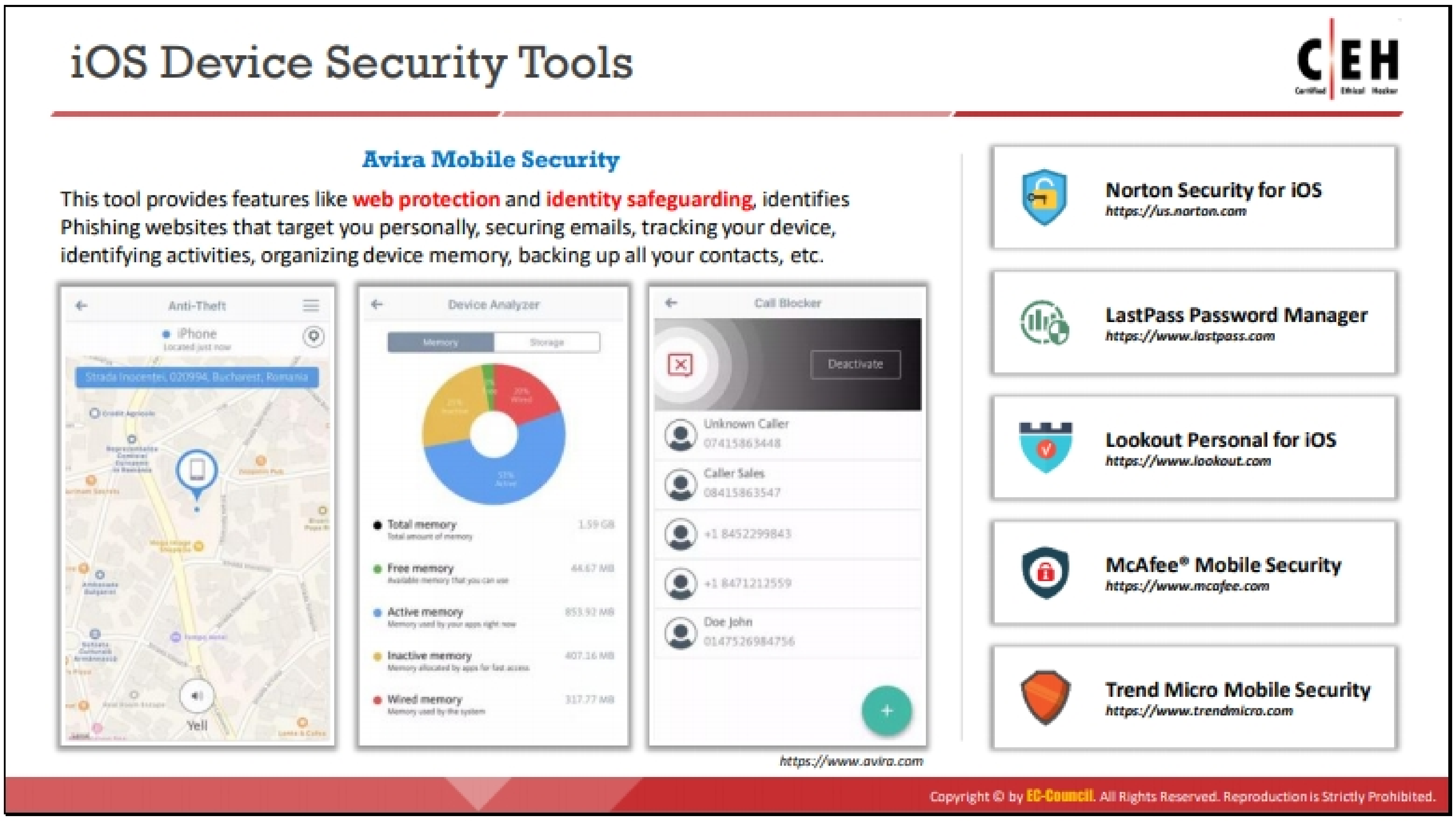
Listed below are some important guidelines to help you secure your iOS devices and their data from attackers:

- Enable the **Passcode Lock** feature on your iPhone. Go to **Settings → Touch ID & Passcode Lock**, and then tap **Turn Passcode On**
- Set separate passcodes for applications containing sensitive data
- Disable Javascript and add-ons from the web browser
- Always download applications from the **Apple App Store**
- Set the **Auto-Lock Timeout** to enter a passcode after a set time. Go to **Settings → General → Auto-Lock**
- Use iOS devices on a secured and protected Wi-Fi network
- Do not store sensitive data on a client-side database
- Do not access web services on a compromised network
- Do not open links or attachments from unknown sources
- Deploy only trusted third-party applications on iOS devices
- Change the default iPhone root password from alpine
- Do not jailbreak or root your device if used within enterprise environments
- Configure Find My iPhone and use it to wipe a lost or stolen device

- Enable Jailbreak detection and also protect access to AppleID and Google accounts, which are tied to sensitive data
- Disable iCloud services so that sensitive enterprise data are not backed up to the cloud (note that cloud services can backup documents, account information, settings, and messages)
- Enable **Ask to Join Networks** function; this prevents you from randomly connecting to available Wi-Fi networks. Go to **Settings → Wi-Fi → Ask to Join Networks**
- Regularly update your device OS with security patches released by Apple. To receive updates, connect to the App Store. For iOS 5 and higher, updates can be received via **Settings → General → Software Updates**
- Enable the Erase Data feature on your iPhone to erase all the data and settings after 10 attempts. Go to **Settings → Touch ID & Passcode → Erase Data**
- Disable the **Voice Dial** feature on iPhone to prevent attackers from accessing the phone without entering a passcode. Go to **Settings → Touch ID & Passcode**, and then **Turn Voice Dial** to **OFF**
- Delete **Keyboard Cache** on your iPhone to remove all your keystrokes recorded. Go to **General → Reset**, tap on **Reset Keyboard Dictionary**, and then **Confirm** on the warning screen
- Disable **Geotagging** (storage of location-based data in images) on the iPhone. Go to **Settings → Privacy → Location Services**, and then toggle the **Camera** to **OFF**
- Enable **Safari's Privacy and Security Settings** on the iPhone. Go to **Settings → Safari**. Here, you can do the following: Enable Block Pop-ups, Disable Passwords and AutoFill, Enable Fraudulent Website Warning, Block cookies, Clear History and Website data, etc.
- Enable the **Do Not Track** feature to keep your web browsing private. Go to **Settings → Safari →** and then enable **Do Not Track** option
- Disable Bluetooth when not in use. Go to **Settings → Bluetooth**, and then toggle it to **OFF**
- Disable Wi-Fi when not in use. Go to **Settings → Wi-Fi**, and then toggle it to **OFF**
- Disable Apple's personal assistant Siri. Go to **Settings → Touch ID & Passcode** and then toggle "**Allow access when locked**" to **OFF**.
- Disable the autofill option in Safari. Go to **Settings → Safari → AutoFill** and then toggle it to **OFF**
- Use two-factor authentication. Go to **Settings → Your Apple ID → Password & Security**, enter your password, and then toggle "**Turn Two-Factor Authentication**" to **ON**
- Install VPN software to encrypt all your Internet traffic
- Install Vault apps to hide the critical data stored on your iOS mobile device

- Reset the connections by navigating to **Settings → General → Reset → Reset location & Privacy** if any suspicious activities are found

Note: The paths given above to enable/disable the respective features may vary based on the iOS version or device used.



iOS Device Security Tools

Avira Mobile Security

This tool provides features like **web protection** and **identity safeguarding**, identifies Phishing websites that target you personally, securing emails, tracking your device, identifying activities, organizing device memory, backing up all your contacts, etc.

Anti-Theft
Device Analyzer
Call Blocker

https://www.avira.com

- **Norton Security for iOS** https://us.norton.com
- **LastPass Password Manager** https://www.lastpass.com
- **Lookout Personal for iOS** https://www.lookout.com
- **McAfee® Mobile Security** https://www.mcafee.com
- **Trend Micro Mobile Security** https://www.trendmicro.com

iOS Device Security Tools

- **Avira Mobile Security**

  Source: *https://www.avira.com*

  The Avira Mobile Security tool provides features such as web protection, identity safeguarding, detection of phishing websites that target you personally, securing emails, tracking your device, identifying activities, organizing device memory, backing up all your contacts, and so on for all iOS devices.

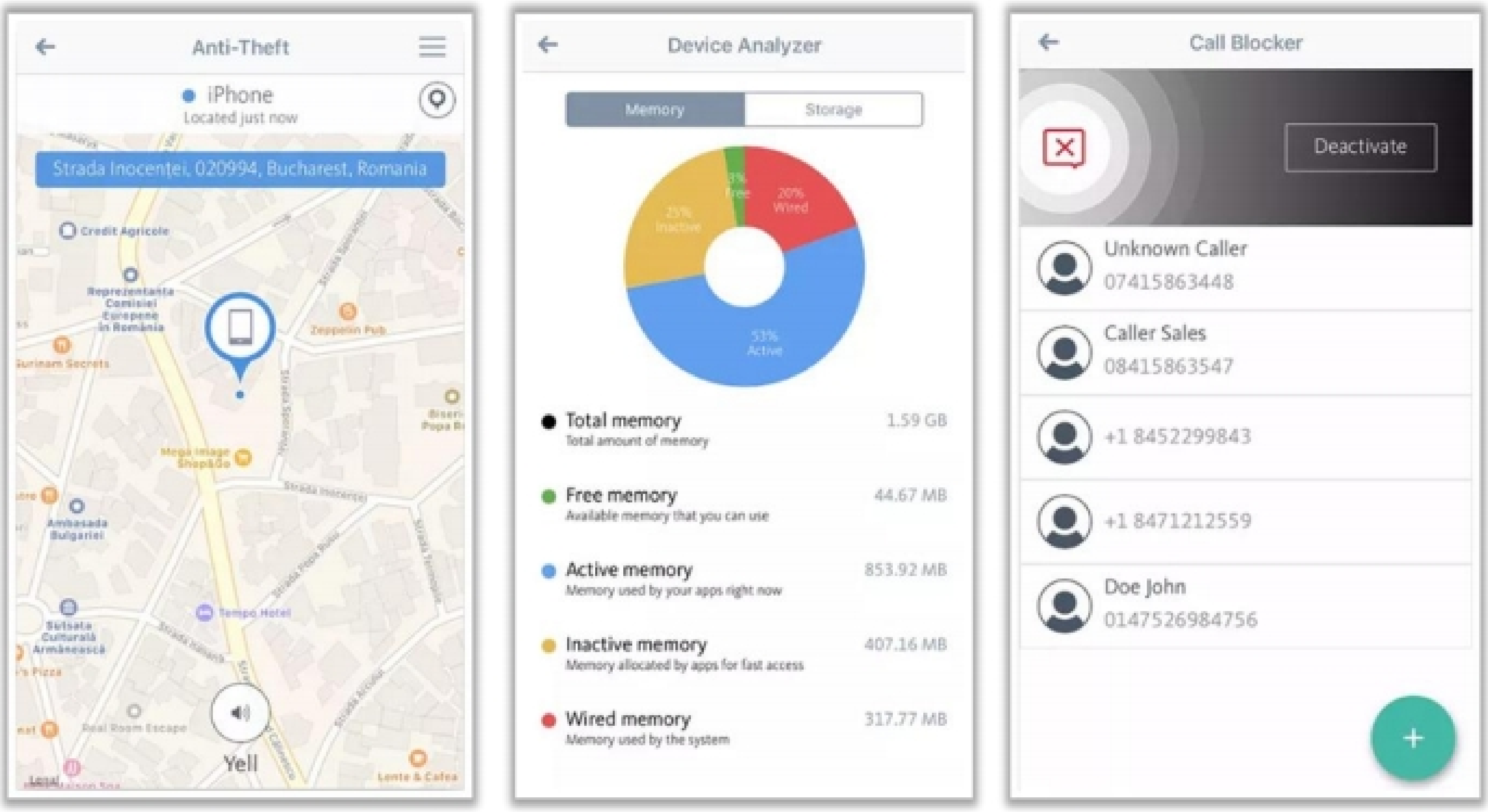
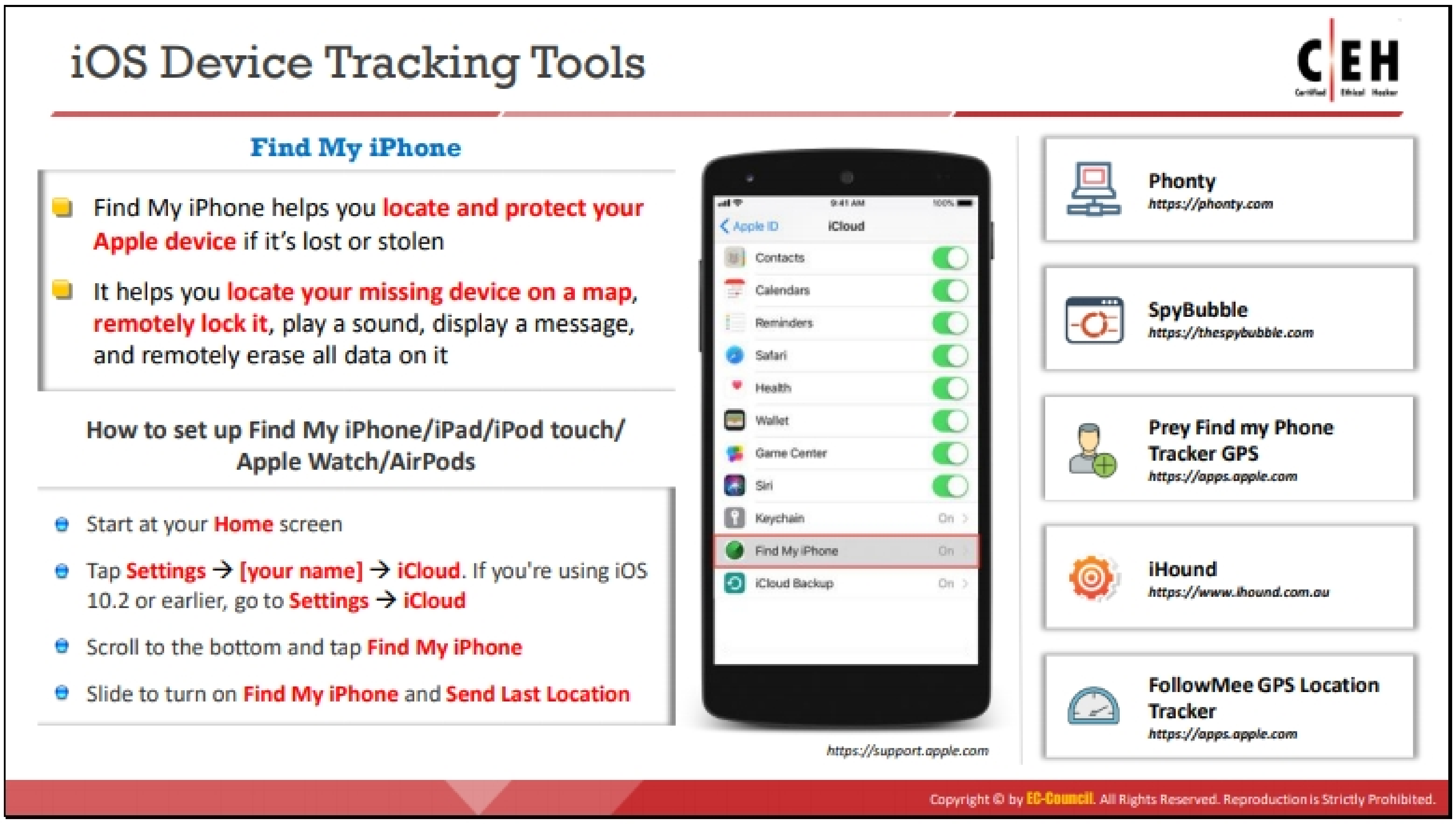

Anti-Theft
iPhone
Located just now
Strada Inocenţei, 020994, Bucharest, Romania
Yell
Device Analyzer
Memory
Storage
3% Free
20% Wired
25% Inactive
53% Active
Total memory
Total amount of memory
1.59 GB
Free memory
Available memory that you can use
44.67 MB
Active memory
Memory used by your apps right now
853.92 MB
Inactive memory
Memory allocated by apps for fast access
407.16 MB
Wired memory
Memory used by the system
317.77 MB
Call Blocker
Deactivate
Unknown Caller
07415863448
Caller Sales
08415863547
+1 8452299843
+1 8471212559
Doe John
0147526984756
+

Figure 17.63: Screenshots of Avira Mobile Security

Some additional iOS device security tools are as follows:

- Norton Security for iOS (*https://us.norton.com*)
- LastPass Password Manager (*https://www.lastpass.com*)
- Lookout Personal for iOS (*https://www.lookout.com*)
- McAfee® Mobile Security (*https://www.mcafee.com*)
- Trend Micro Mobile Security (*https://www.trendmicro.com*)



iOS Device Tracking Tools

Find My iPhone

- Find My iPhone helps you **locate and protect your Apple device** if it's lost or stolen
- It helps you **locate your missing device on a map**, **remotely lock it**, play a sound, display a message, and remotely erase all data on it

How to set up Find My iPhone/iPad/iPod touch/ Apple Watch/AirPods

- Start at your **Home** screen
- Tap **Settings** → **[your name]** → **iCloud**. If you're using iOS 10.2 or earlier, go to **Settings** → **iCloud**
- Scroll to the bottom and tap **Find My iPhone**
- Slide to turn on **Find My iPhone** and **Send Last Location**

https://support.apple.com

Phonty
https://phonty.com

SpyBubble
https://thespybubble.com

Prey Find my Phone Tracker GPS
https://apps.apple.com

iHound
https://www.ihound.com.au

FollowMee GPS Location Tracker
https://apps.apple.com

iOS Device Tracking Tools

Some iOS device tracking tools are listed below:

- **Find My iPhone**

  Source: *https://support.apple.com*

  Find My iPhone is an iOS device tracking tool that allows you to use another iOS device to track a lost or misplaced iPhone, iPad, iPod touch, or Mac and protect their data. To use this tool, you need to install the app on another iOS device, open it, and sign in with your Apple ID. It helps you locate your missing device on a map, remotely lock it, play a sound, display a message, and erase all the data on it.

  Find My iPhone also includes Lost Mode to locate an iDevice running iOS 6 or higher. Lost Mode locks your missing device with a passcode and displays a custom message such as a contact phone number on the lock screen. While in Lost Mode, your device also keeps track of where it has been so that you can view its recent location history from the Find My iPhone app.

  How to set up Find My iPhone, iPad, iPod touch, Apple Watch, AirPods

  1. Start at your **Home** screen
  2. Tap **Settings** → **[your name]** → **iCloud**. If you are using iOS 10.2 or lower, go to **Settings** → **iCloud**
  3. Scroll to the bottom and tap **Find My iPhone**
  4. Slide to turn on **Find My iPhone** and **Send Last Location**

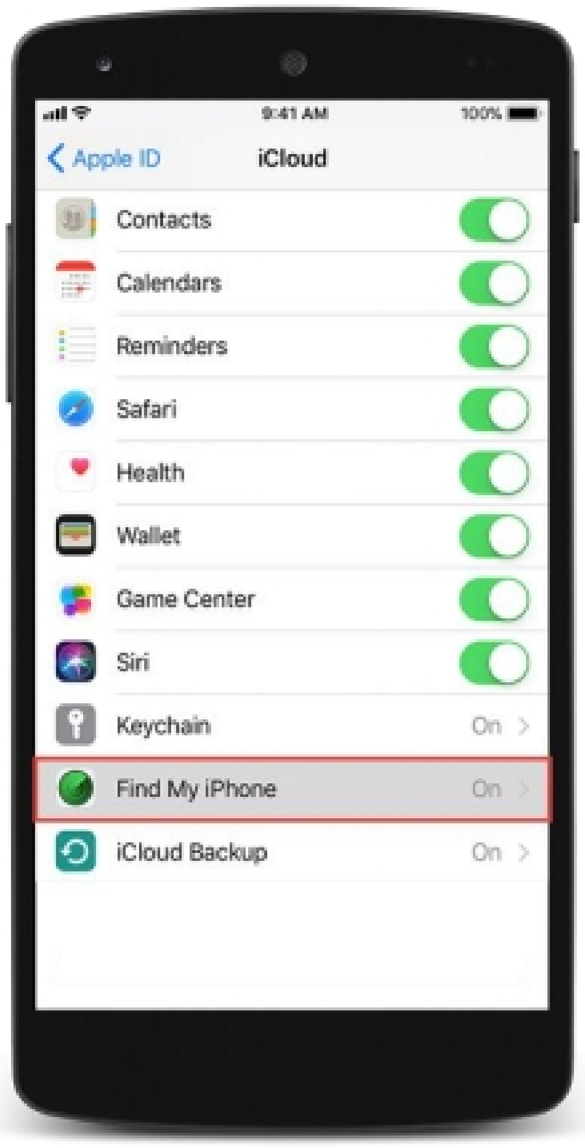
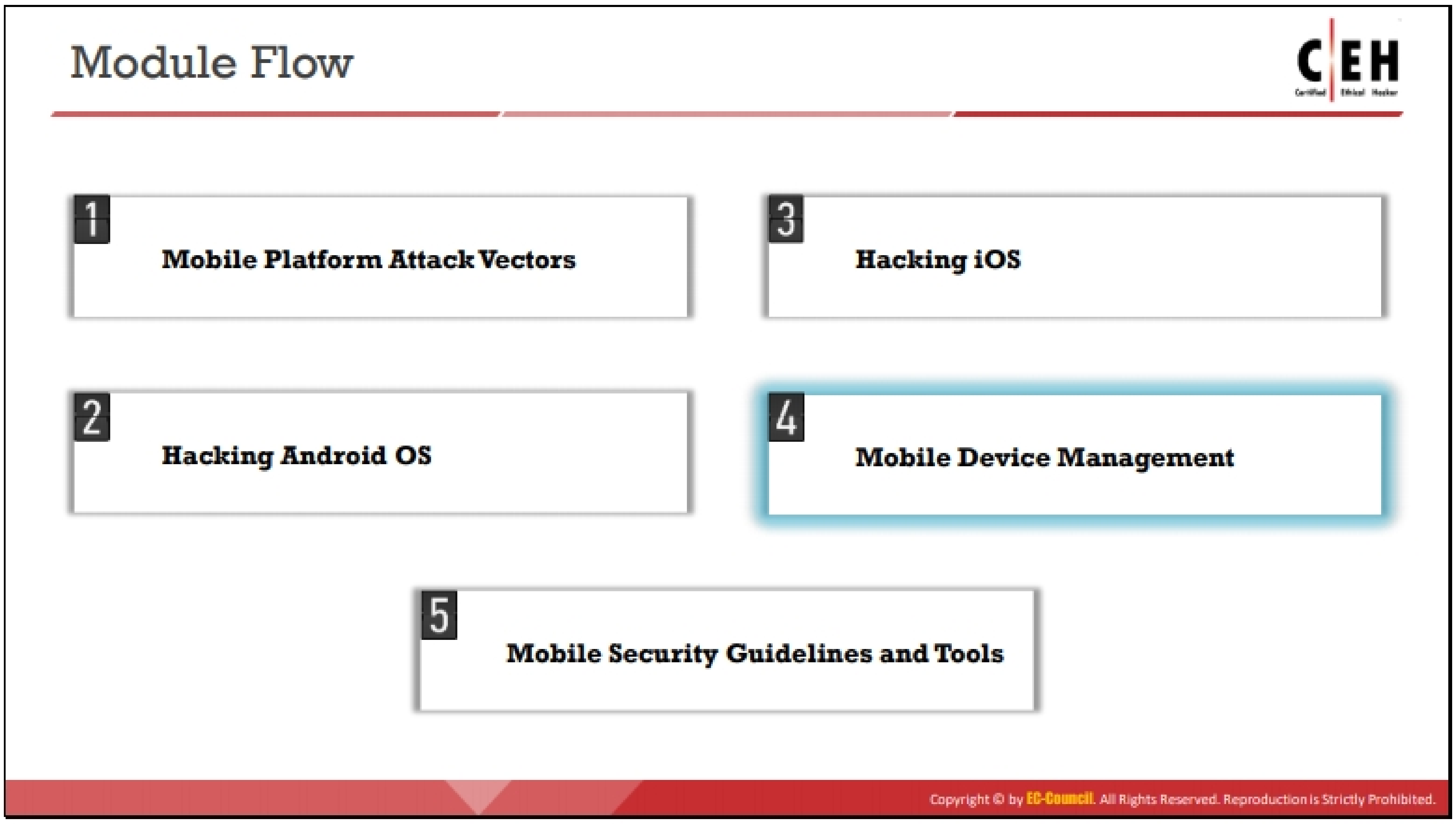

9:41 AM
100%
Apple ID
iCloud
Contacts
Calendars
Reminders
Safari
Health
Wallet
Game Center
Siri
Keychain
On
Find My iPhone
On
iCloud Backup
On

Figure 17.64: Screenshot of Find My iPhone

Some additional iOS device tracking tools are as follows:

- Phonty (*https://phonty.com*)
- SpyBubble (*https://thespybubble.com*)
- Prey Find my Phone Tracker GPS (*https://apps.apple.com*)
- iHound (*http://ihoundgps.com*)
- FollowMee GPS Location Tracker (*https://apps.apple.com*)



Module Flow

1 **Mobile Platform Attack Vectors**

2 **Hacking Android OS**

3 **Hacking iOS**

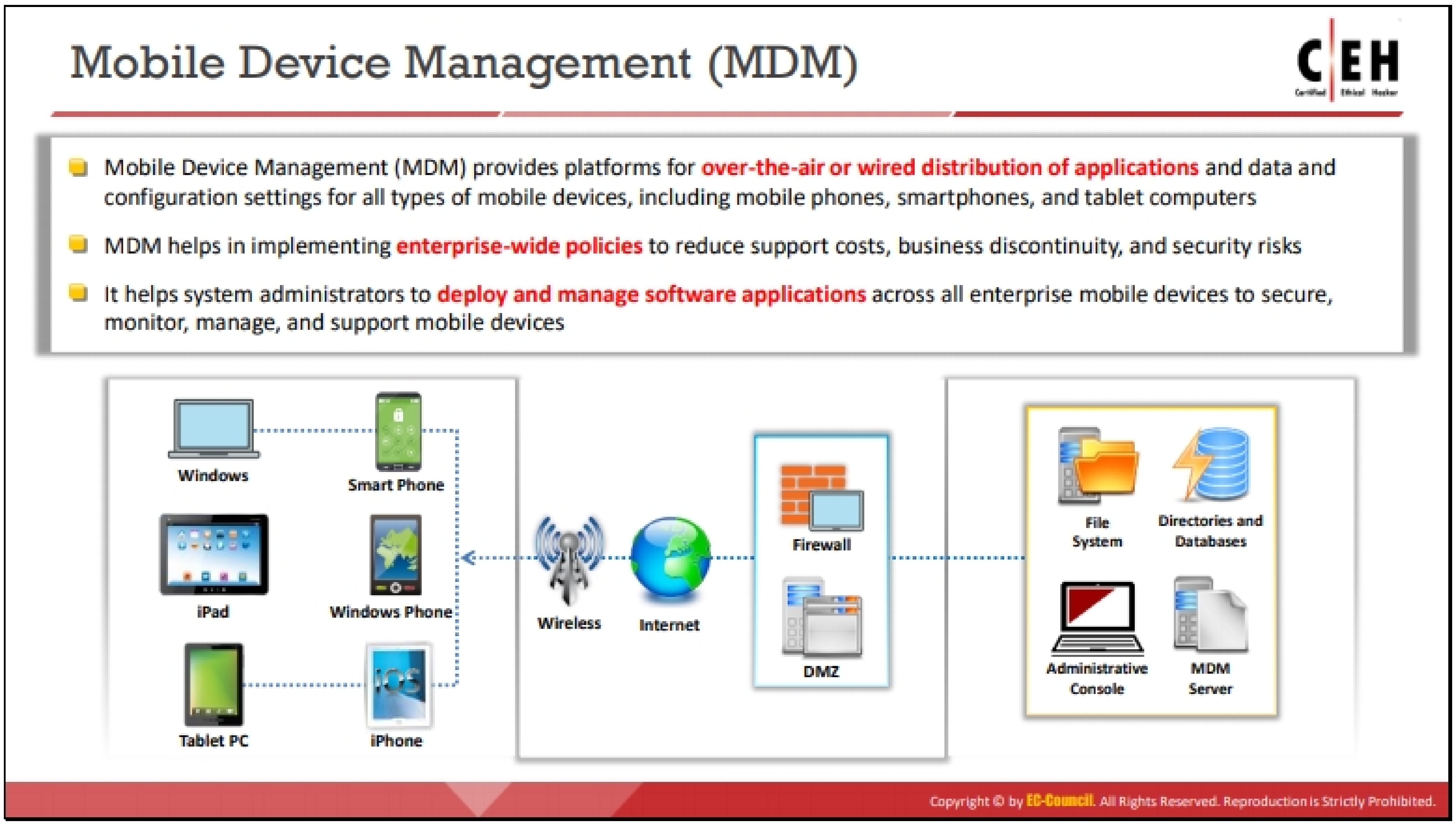
4 **Mobile Device Management**

5 **Mobile Security Guidelines and Tools**

Mobile Device Management

Mobile device management (MDM) is gaining considerable importance with the adoption of policies such as BYOD across organizations. The increasing number and types of mobile devices such as smartphones, laptops, tablets, and so on has made it difficult for enterprises to make policies and manage these devices securely. MDM is a policy that helps to handle such devices carefully while ensuring that they are secure. Companies use a kind of security software for the administration of all mobile devices connected to the enterprise network.

This section deals with MDM and its solutions that help to secure, monitor, manage, and support mobile devices.



Mobile Device Management (MDM)

- Mobile Device Management (MDM) provides platforms for **over-the-air or wired distribution of applications** and data and configuration settings for all types of mobile devices, including mobile phones, smartphones, and tablet computers
- MDM helps in implementing **enterprise-wide policies** to reduce support costs, business discontinuity, and security risks
- It helps system administrators to **deploy and manage software applications** across all enterprise mobile devices to secure, monitor, manage, and support mobile devices

Mobile Device Management (MDM)

MDM provides platforms for over-the-air or wired distribution of applications, data, and configuration settings for all types of mobile devices, including mobile phones, smartphones, tablet computers, and so on. It helps in implementing enterprise-wide policies to reduce support costs, business discontinuity, and security risks. It helps system administrators to deploy and manage software applications across all enterprise mobile devices to secure, monitor, manage, and support these devices. It can be used to manage both company-owned and employee-owned (BYOD) devices across the enterprise.

The basic features of MDM software are as follows:

- Uses a passcode for the device
- Remotely locks the device if it is lost
- Remotely wipes data in the lost or stolen device
- Detects if the device is rooted or jailbroken
- Enforces policies and tracks inventory
- Performs real-time monitoring and reporting

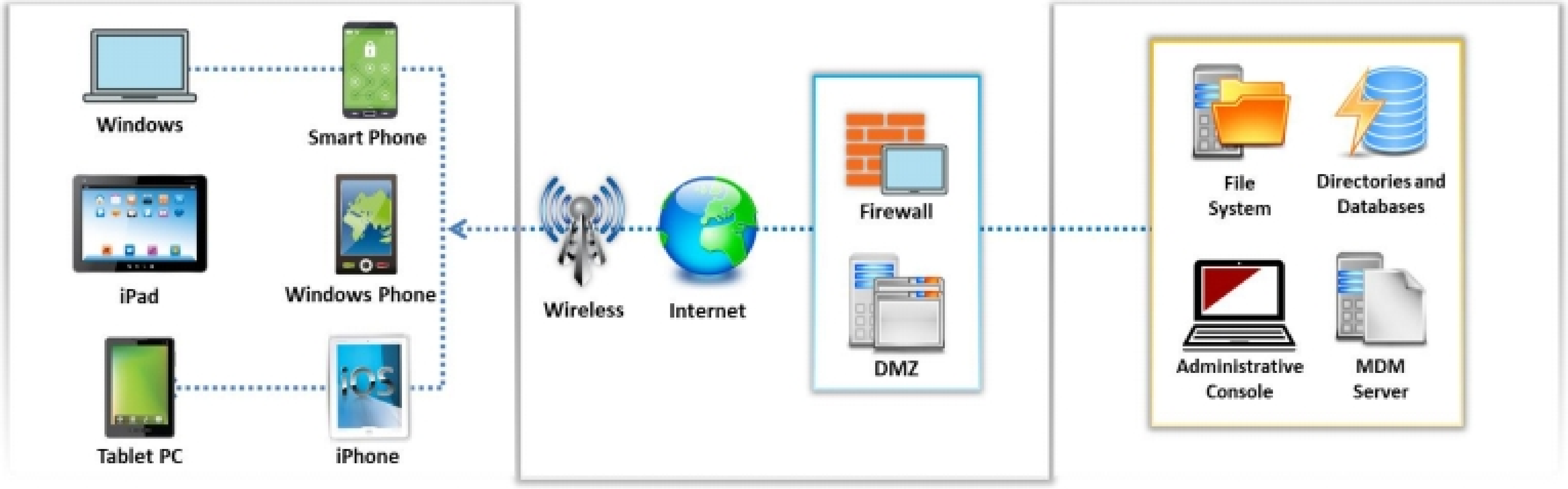
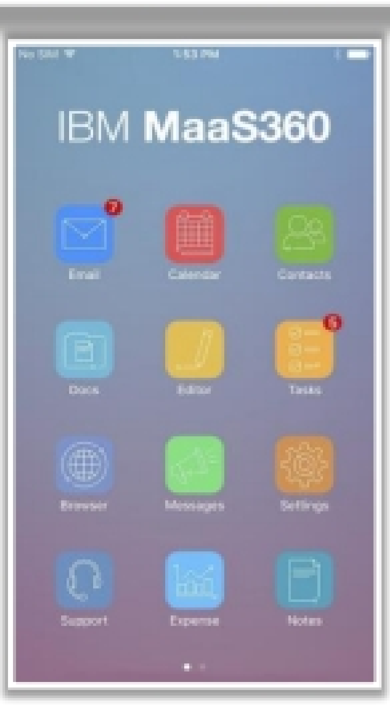
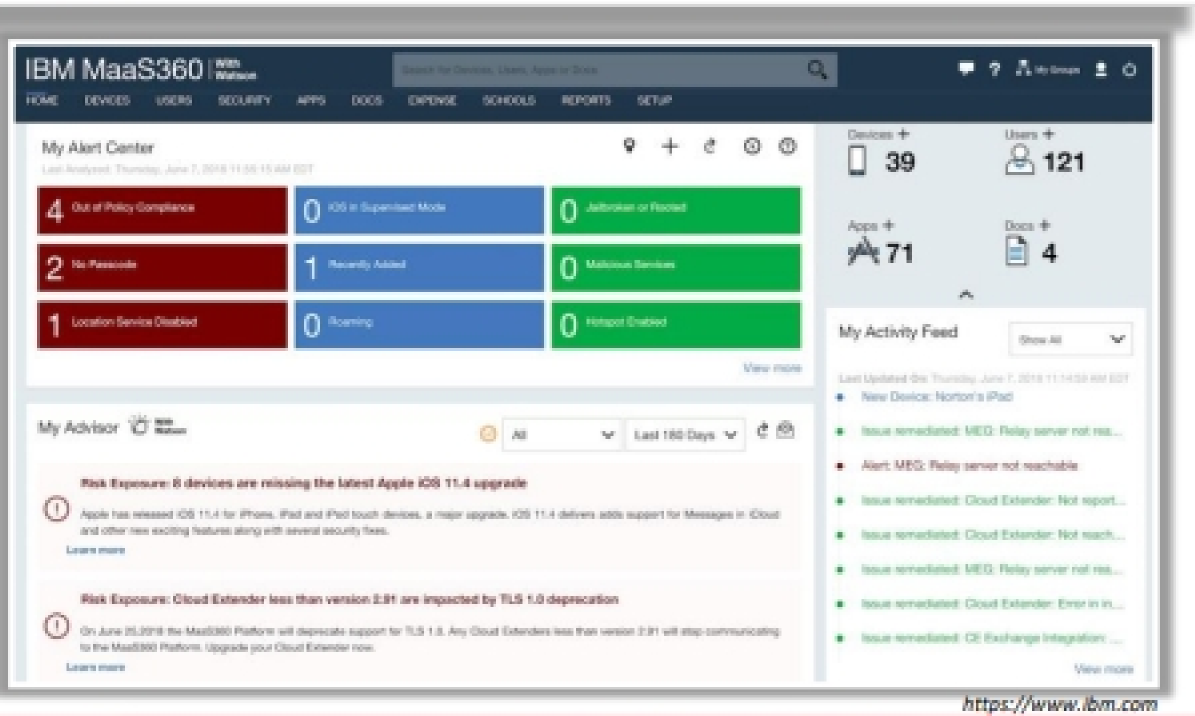

Windows
Smart Phone
iPad
Windows Phone
Tablet PC
iPhone
Wireless
Internet
Firewall
DMZ
File System
Directories and Databases
Administrative Console
MDM Server

Figure 17.65: Schematic of Mobile Device Management (MDM)

Mobile Device Management Solutions: IBM MaaS360

- MaaS360 supports the complete **mobile device management (MDM) lifecycle** for smartphones and tablets, including iPhone, iPad, Android, Windows Phone, and Kindle Fire
- As a **fully integrated cloud platform**, MaaS360 simplifies MDM with rapid deployment and comprehensive visibility and control that spans across mobile devices, applications, and documents


IBM MaaS360
Email
Calendar
Contacts
Docs
Editor
Tasks
Browser
Messages
Settings
Support
Expense
Notes


IBM MaaS360 With Watson
HOME
DEVICES
USERS
SECURITY
APPS
DOCS
EXPENSE
SCHOOLS
REPORTS
SETUP
My Alert Center
4 Out of Policy Compliance
0 iOS in Supervised Mode
0 Jailbroken or Rooted
2 No Passcode
1 Recently Added
0 Malicious Devices
1 Location Service Disabled
0 Roaming
0 Hotspot Enabled
View more
Devices 39
Users 121
Apps 71
Docs 4
My Activity Feed
My Advisor
All
Last 180 Days
View more

Risk Exposure: 8 devices are missing the latest Apple iOS 11.4 upgrade

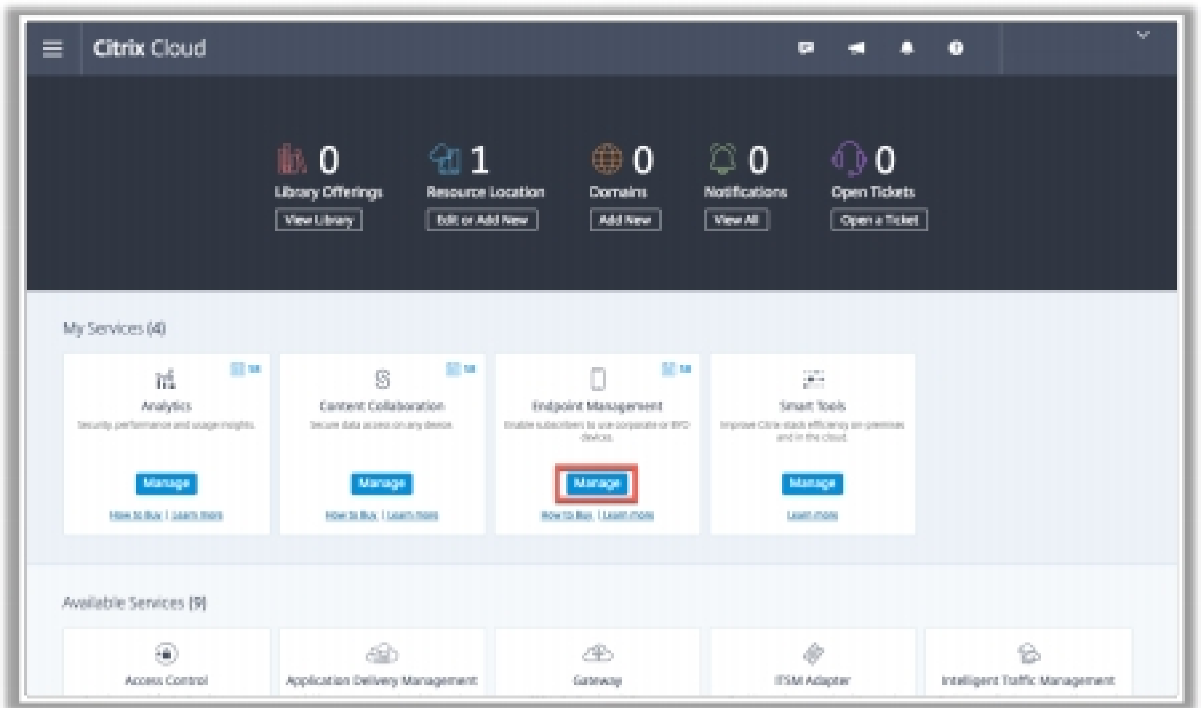
Risk Exposure: Cloud Extender less than version 2.91 are impacted by TLS 1.0 deprecation

https://www.ibm.com

Mobile Device Management Solutions

Citrix Endpoint Management

Citrix Endpoint Management provides a modern approach to manage various devices, including desktops, laptops, smartphones, tablets, and IoT, through a single platform


Citrix Cloud
0 Library Offerings
View Library
1 Resource Location
Edit or Add New
0 Domains
Add New
0 Notifications
View All
0 Open Tickets
Open a Ticket
My Services (4)
Analytics
Content Collaboration
Endpoint Management
Smart Tools
Manage
Manage
Manage
Manage
Available Services (9)
Access Control
Application Delivery Management
Gateway
ITSM Adapter
Intelligent Traffic Management

https://www.citrix.com

VMware AirWatch
https://www.air-watch.com

Sicap Device Management Centre
https://www.sicap.com

SOTI MobiControl
https://www.soti.net

Scalefusion MDM
https://scalefusion.com

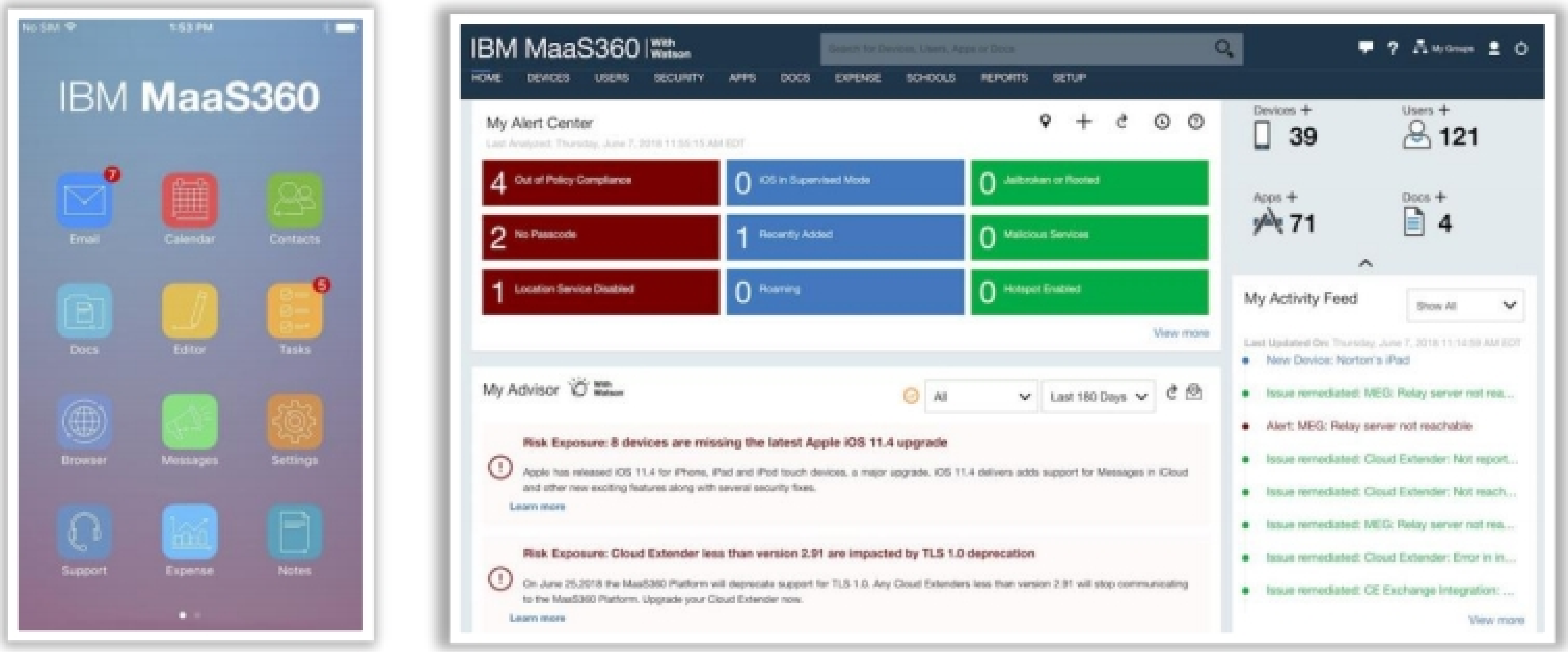
ManageEngine Mobile Device Manager Plus
https://www.manageengine.com

Mobile Device Management Solutions

- **IBM MaaS360**

  Source: *https://www.ibm.com*

  MaaS360 supports the complete MDM lifecycle for smartphones and tablets including iPhone, iPad, Android, Windows Phone, and Kindle Fire. As a fully integrated cloud platform, MaaS360 simplifies MDM with rapid deployment as well as comprehensive visibility and control that spans across mobile devices, applications, and documents. IBM MaaS360 helps security professionals to rapidly enroll mobile devices, integrate mobile devices with enterprise systems, and centrally manage and secure mobile devices.


IBM MaaS360
Email
Calendar
Contacts
Docs
Editor
Tasks
Browser
Messages
Settings
Support
Expense
Notes
IBM MaaS360 With Watson
HOME
DEVICES
USERS
SECURITY
APPS
DOCS
EXPENSE
SCHOOLS
REPORTS
SETUP
My Alert Center
4 Out of Policy Compliance
0 iOS in Supervised Mode
0 Jailbroken or Rooted
2 No Passcode
1 Recently Added
0 Malicious Services
1 Location Service Disabled
0 Roaming
0 Hotspot Enabled
View more
Devices 39
Users 121
Apps 71
Docs 4
My Activity Feed
Show All
New Device: Norton's iPad
Issue remediated: MEG: Relay server not rea...
Alert: MEG: Relay server not reachable
Issue remediated: Cloud Extender: Not report...
Issue remediated: Cloud Extender: Not reach...
Issue remediated: MEG: Relay server not rea...
Issue remediated: Cloud Extender: Error in in...
Issue remediated: CE Exchange Integration: ...
View more
My Advisor With Watson
All
Last 180 Days
Learn more

Risk Exposure: 8 devices are missing the latest Apple iOS 11.4 upgrade

Apple has released iOS 11.4 for iPhone, iPad and iPod touch devices, a major upgrade. iOS 11.4 delivers adds support for Messages in iCloud and other new exciting features along with several security fixes.

Risk Exposure: Cloud Extender less than version 2.91 are impacted by TLS 1.0 deprecation

On June 25,2018 the MaaS360 Platform will deprecate support for TLS 1.0. Any Cloud Extenders less than version 2.91 will stop communicating to the MaaS360 Platform. Upgrade your Cloud Extender now.

Figure 17.66: Screenshot of IBM Mobile Device Management MaaS360

- **Citrix Endpoint Management**

  Source: *https://www.citrix.com*

  Citrix Endpoint Management provides a modern approach to manage various devices including desktops, laptops, smartphones, tablets, and IoT through a single platform. When deployed as part of Citrix Workspace, it allows users to access all their apps and files from a single, intuitive interface, providing a consistent and smooth experience across any device.

  With Citrix Endpoint Management, organizations can ensure that they have combined policies, procedures, and technologies to protect their corporate data regardless of where it is accessed or resides. Through mobile app and device management, Citrix Endpoint Management secures corporate data and mobilizes productivity, enabling device freedom for end users without the added stress on IT.

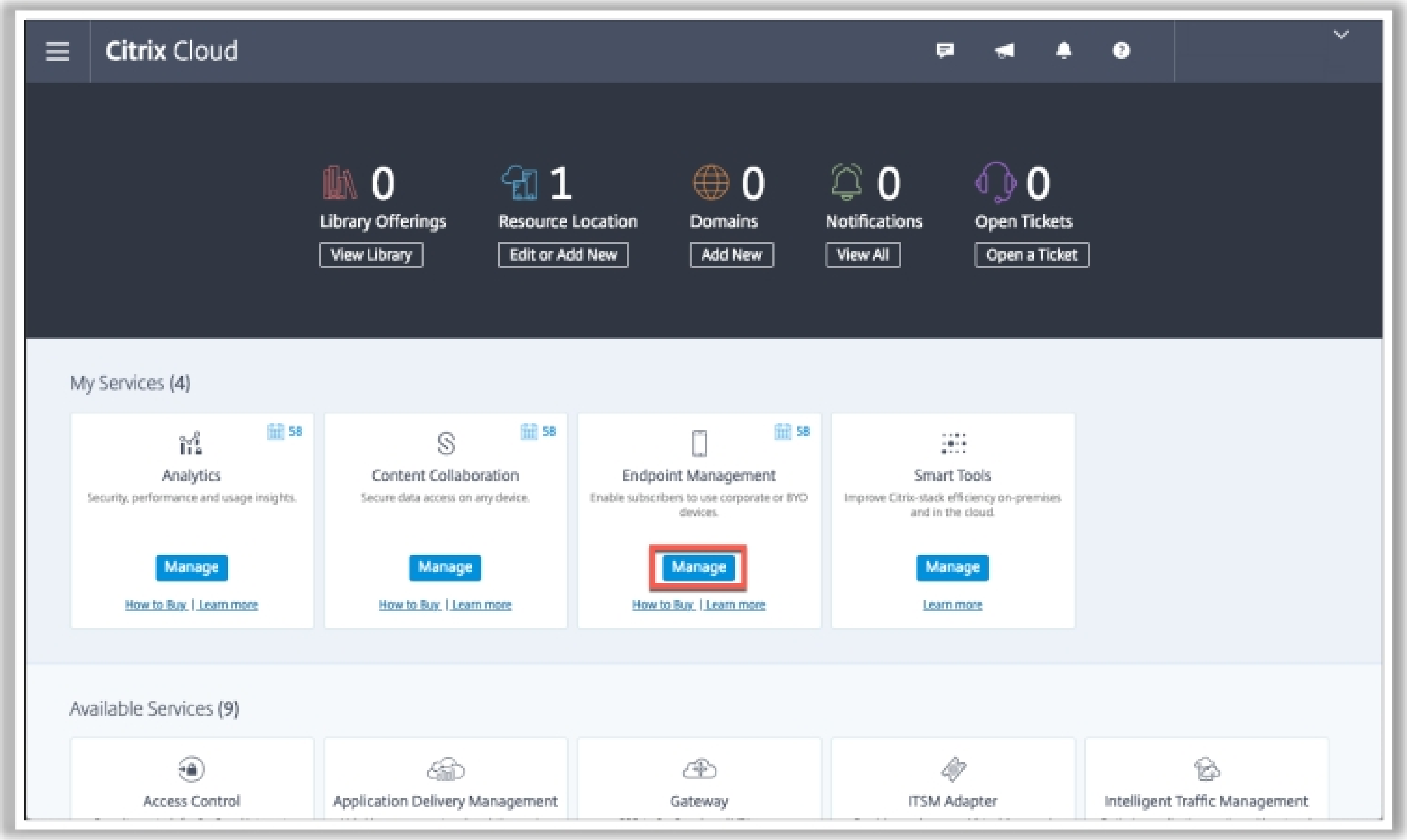
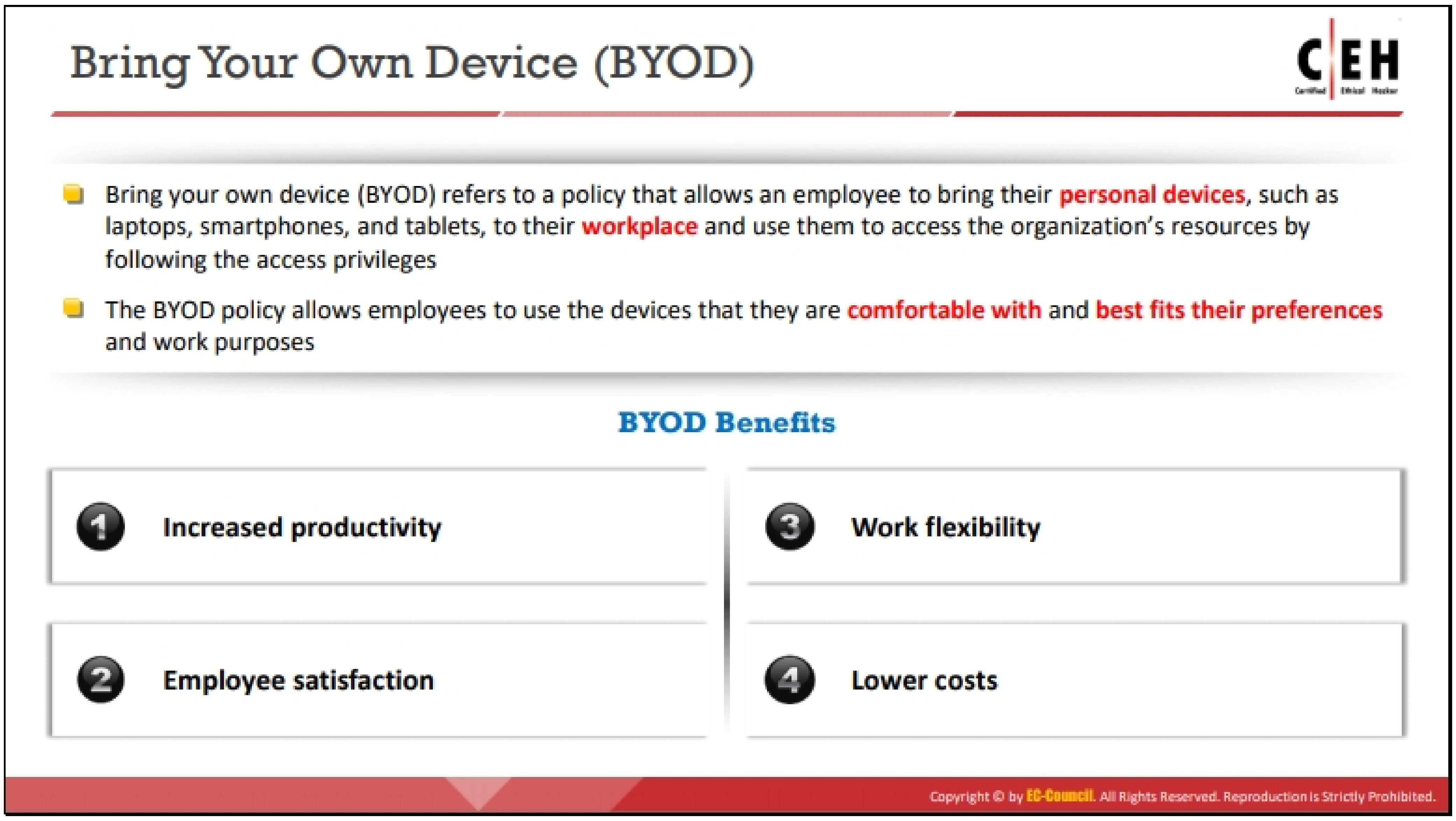

Citrix Cloud
0
Library Offerings
View Library
1
Resource Location
Edit or Add New
0
Domains
Add New
0
Notifications
View All
0
Open Tickets
Open a Ticket
My Services (4)
58
Analytics
Security, performance and usage insights.
Manage
How to Buy | Learn more
58
Content Collaboration
Secure data access on any device.
Manage
How to Buy | Learn more
58
Endpoint Management
Enable subscribers to use corporate or BYO devices.
Manage
How to Buy | Learn more
Smart Tools
Improve Citrix-stack efficiency on-premises and in the cloud.
Manage
Learn more
Available Services (9)
Access Control
Application Delivery Management
Gateway
ITSM Adapter
Intelligent Traffic Management

Figure 17.67: Screenshot of Citrix XenMobile

Some additional MDM solutions are as follows:

- VMware AirWatch (*https://www.air-watch.com*)
- Sicap Device Management Centre (*https://www.sicap.com*)
- SOTI MobiControl (*https://www.soti.net*)
- Scalefusion MDM (*https://scalefusion.com*)
- ManageEngine Mobile Device Manager Plus (*https://www.manageengine.com*)



Bring Your Own Device (BYOD)

- Bring your own device (BYOD) refers to a policy that allows an employee to bring their **personal devices**, such as laptops, smartphones, and tablets, to their **workplace** and use them to access the organization's resources by following the access privileges
- The BYOD policy allows employees to use the devices that they are **comfortable with** and **best fits their preferences** and work purposes

BYOD Benefits

1. **Increased productivity**
2. **Employee satisfaction**
3. **Work flexibility**
4. **Lower costs**

Bring Your Own Device (BYOD)

BYOD refers to a policy that allows employees to bring their personal devices such as laptops, smartphones, and tablets to their workplace and use them for accessing the organization's resources as per their access privileges.

BYOD allows employees to use devices that they are comfortable with and which best fits their preferences and work purposes. With a "work anywhere, anytime" strategy, the challenge for the BYOD trend is to secure the company's data and meet compliance requirements.

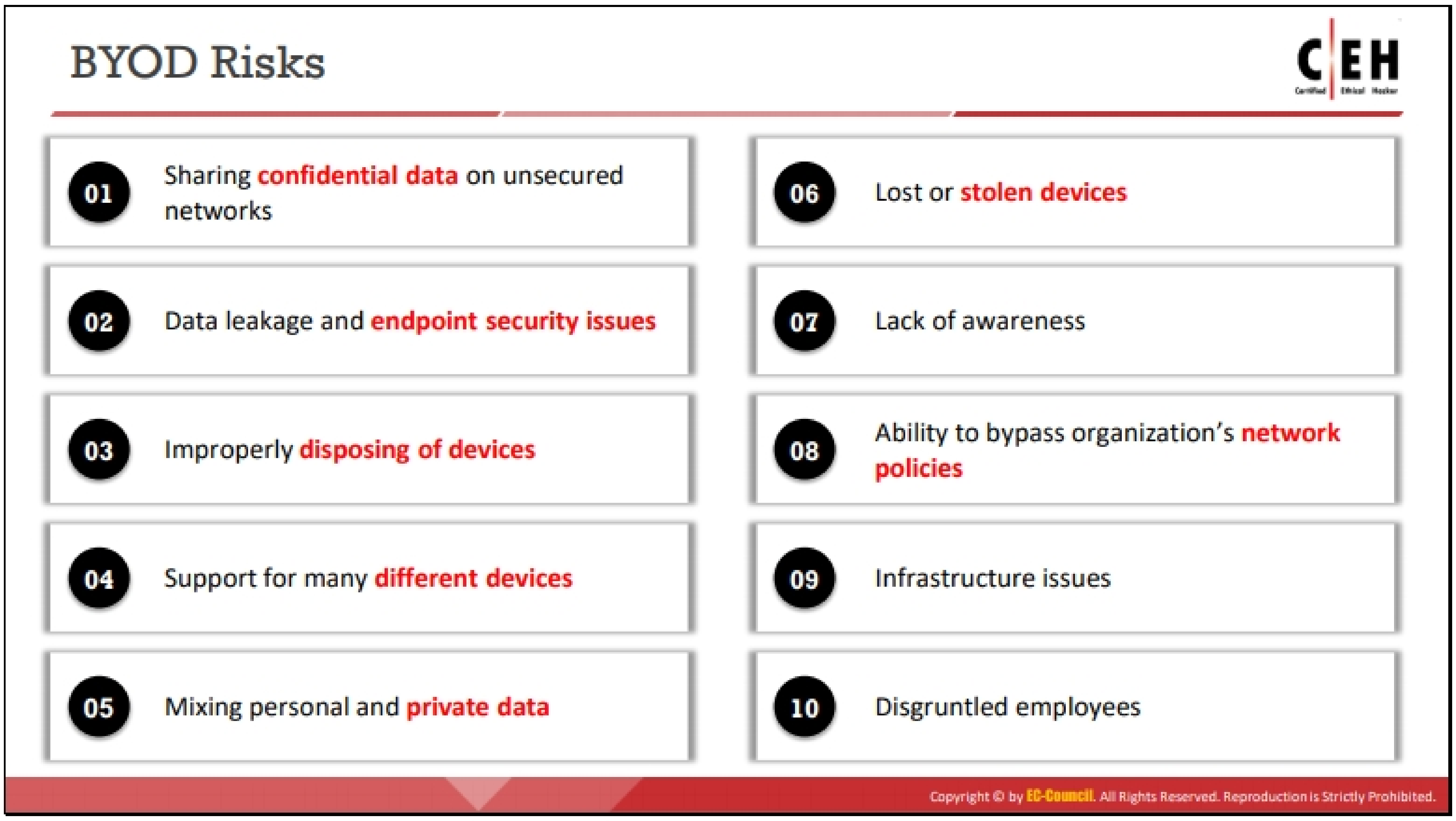
BYOD Benefits

Adopting BYOD advantageous for the company as well as the employee. Some of the benefits of BYOD are discussed below:

- **Increased Productivity**: Employees become experts in using their personal devices and this increases their productivity. In addition, users tend to upgrade their personal devices with cutting-edge technologies so that the enterprise can benefit from the latest features (both software and hardware) of the device.
- **Employee Satisfaction**: By implementing BYOD, employees use devices of their own choice, which they invest in themselves without the company's involvement. Moreover, employees are more comfortable with their personal devices, as they contain both personal data and corporate data, thus eliminating the usage of multiple devices.
- **Work Flexibility**: By practicing BYOD, employees can carry a single device to satisfy their personal and professional needs. Work that is usually done in the office can be done from anywhere in the world, as employees are provided with access to the corporate data. BYOD users have more freedom, as their companies do not impose strict rules that they would have to follow when using company property. BYOD replaces the traditional

client-server model with a mobile and cloud-centric strategy, which can have far-reaching benefits.

- **Lower Costs:** A business that adopts BYOD does not have to spend on devices but saves money, as employees purchase their own devices. In addition, the cost of data services shifts to employees who can take better care of their own property (device).



BYOD Risks

- 01 Sharing confidential data on unsecured networks
- 02 Data leakage and endpoint security issues
- 03 Improperly disposing of devices
- 04 Support for many different devices
- 05 Mixing personal and private data
- 06 Lost or stolen devices
- 07 Lack of awareness
- 08 Ability to bypass organization's network policies
- 09 Infrastructure issues
- 10 Disgruntled employees

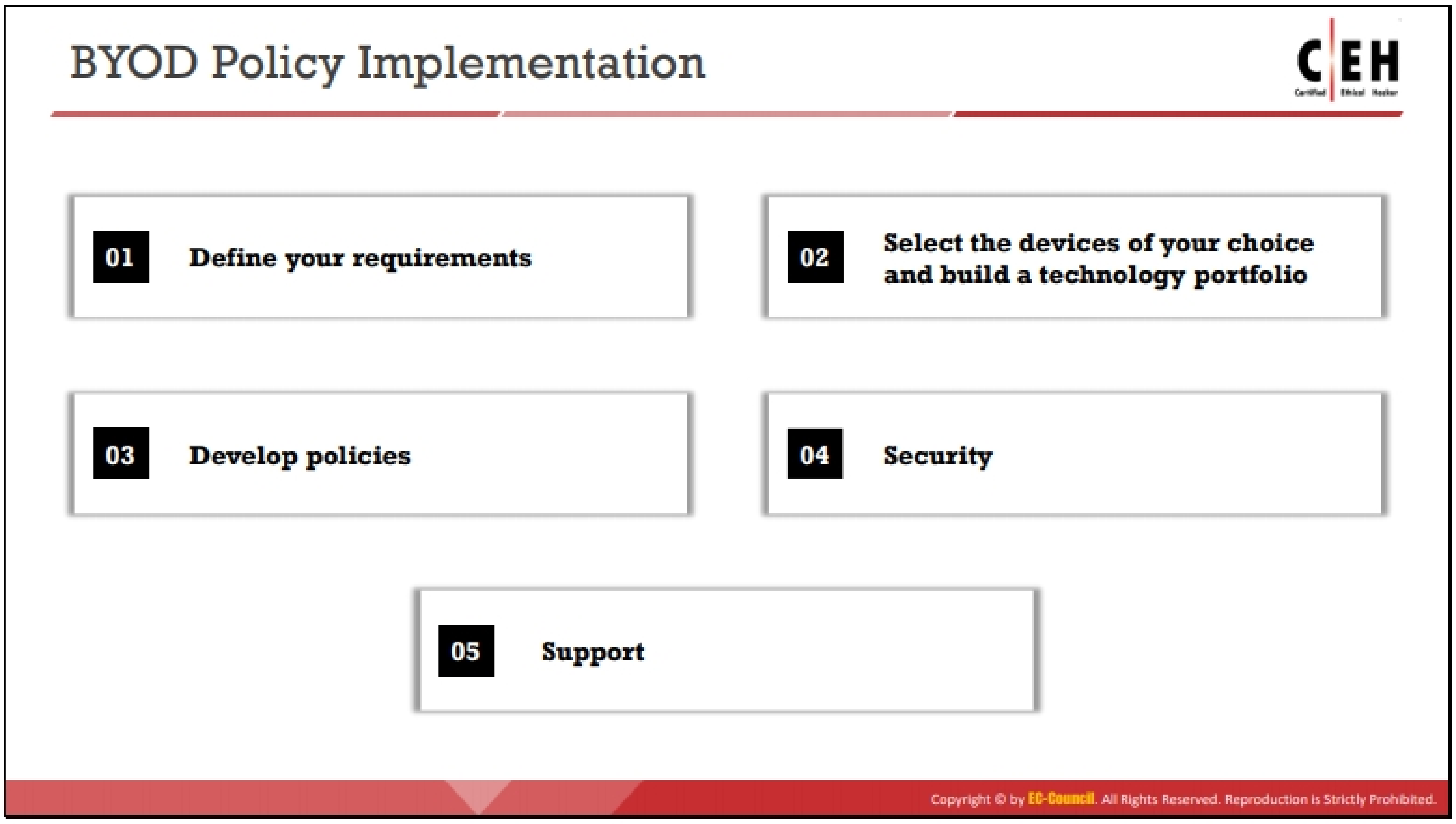
BYOD Risks

Employees connecting to the corporate network or accessing corporate data using their own mobile devices pose security risks to the organization. Some BYOD security risks are listed below:

- **Sharing confidential data on unsecured networks**: Employees might access corporate data via a public network. These connections may not be encrypted; sharing confidential data via an unsecured network may lead to data leakage.
- **Data leakage and endpoint security issues**: In this cloud-computing era, mobile devices are insecure endpoints with cloud connectivity. By synchronizing with organizational email or other apps, these mobile devices carry confidential information. If the device is lost, it could potentially expose all the corporate data.
- **Improperly disposing of devices**: An improperly disposed of device could contain a wealth of sensitive information, such as financial information, credit card details, contact numbers, and corporate data. Therefore, it is important to ensure that the device does not contain any data before it is disposed of or passed on to others.
- **Support for many different devices**: Organizations allow employees to access its resources from anywhere in the world, enhancing productivity and driving employee satisfaction. However, support for different devices and processes can increase costs. Employee-owned devices have limited security and come with a variety of platforms. This impedes the IT department's capability to manage and control all the devices in a company.
- **Mixing personal and private data**: Mixing personal and corporate data on mobile devices leads to serious security and privacy implications. Therefore, it is a good practice

to keep the corporate data separate from the employee's personal data; this helps an organization to apply specific security measures such as encryption to protect the critical corporate data stored on the mobile device. In addition, it becomes easy for the organization to remotely wipe the corporate data without affecting the employee's personal data when an employee leaves the organization.

- **Lost or stolen devices**: Due to their small size, mobile devices are often lost or stolen. When an employee loses his/her mobile device that is used for both personal and official purposes, the organization might face a security risk, as attackers can compromise the corporate data stored in the lost device.
- **Lack of awareness**: Organizations must educate their employees regarding BYOD security issues. Failing to do so might result in compromising the corporate data stored in mobile devices.
- **Ability to bypass organization's network policy rules**: According to their particular requirements, the policies imposed may differ between wired networks and wireless networks. BYOD devices connected to wireless networks have the ability to bypass the organization's network policy rules enforced only on wired LANs.
- **Infrastructure issues**: A BYOD program involves dealing with various platforms and technologies. Not all employees carry the same devices. Different devices, each running different OS and programs, come with their own security loopholes. Thus, it can be problematic for an IT department to set up and maintain infrastructure to support different devices' needs, such as managing data, security, backup, and compatibility among devices.
- **Disgruntled employees**: Disgruntled employees in an organization can misuse corporate data stored on their mobile devices. They may also leak sensitive information to competitors.



BYOD Policy Implementation

01 **Define your requirements**

02 **Select the devices of your choice and build a technology portfolio**

03 **Develop policies**

04 **Security**

05 **Support**

BYOD Policy Implementation

It could be argued that an organization could reap significant benefits through BYOD policy implementation, ranging from higher user satisfaction to greater productivity due to working with advanced devices. However, the nature of new technology and processes could pose risks to an organization if they are not properly managed.

The five principles involved in BYOD policy implementation are discussed below. Using these principles, an organization can minimize risks associated with data security and privacy.

- **Define your requirements**

  Not all user requirements are alike. Thus, organize or group employees using mobile devices at work into segments considering job criticality, time sensitivity, value derived from mobility, data access, and systems access. It is best to define end user segments by the location/type of worker (e.g., employee working from home, full-time remote, day extender, part-time remote). Next, assign a technology portfolio for each segment, as per user needs.

  Perform a privacy impact assessment (PIA) at the very beginning of each BYOD project in the presence of all the relevant teams after assigning responsibilities and collecting the requirements. It provides an organized procedure to document facts, objectives, privacy risks, and risk mitigation approaches and decisions throughout the project life cycle. It should be a central activity performed by your mobile governance committee (end users from each segment/line of business and IT management).

- **Select the devices of your choice and build a technology portfolio**

  Decide how you want to manage your users and their data access. Apart from the MDM system that provides a minimum level of control, you may use other options such as virtual desktops or on-device software to improve security and data privacy. In addition, ensure that your corporate environment supports WLAN device connectivity and management.

- **Develop policies**

  A delegation of company resources (not just IT) should develop the policies. It should include key participants such as HR, legal, security, and privacy.

  The key components of a general BYOD policy are as follows:

  - Information security concerns
  - Data protection concerns
  - Confidentiality and ownership issues
  - Information regarding any tracking/monitoring
  - Considerations regarding the termination of employment
  - Guidance regarding how to assess the security of Wi-Fi networks
  - Acceptable and unacceptable behavior

  Ensure that end users have a clear idea about the acceptable-use policy prior to entering a BYOD program. Finally, organizations must ensure that their BYOD policy is applicable against their employees and any third parties on their behalf, should the need arise, and follow through with its implementation.

- **Security**

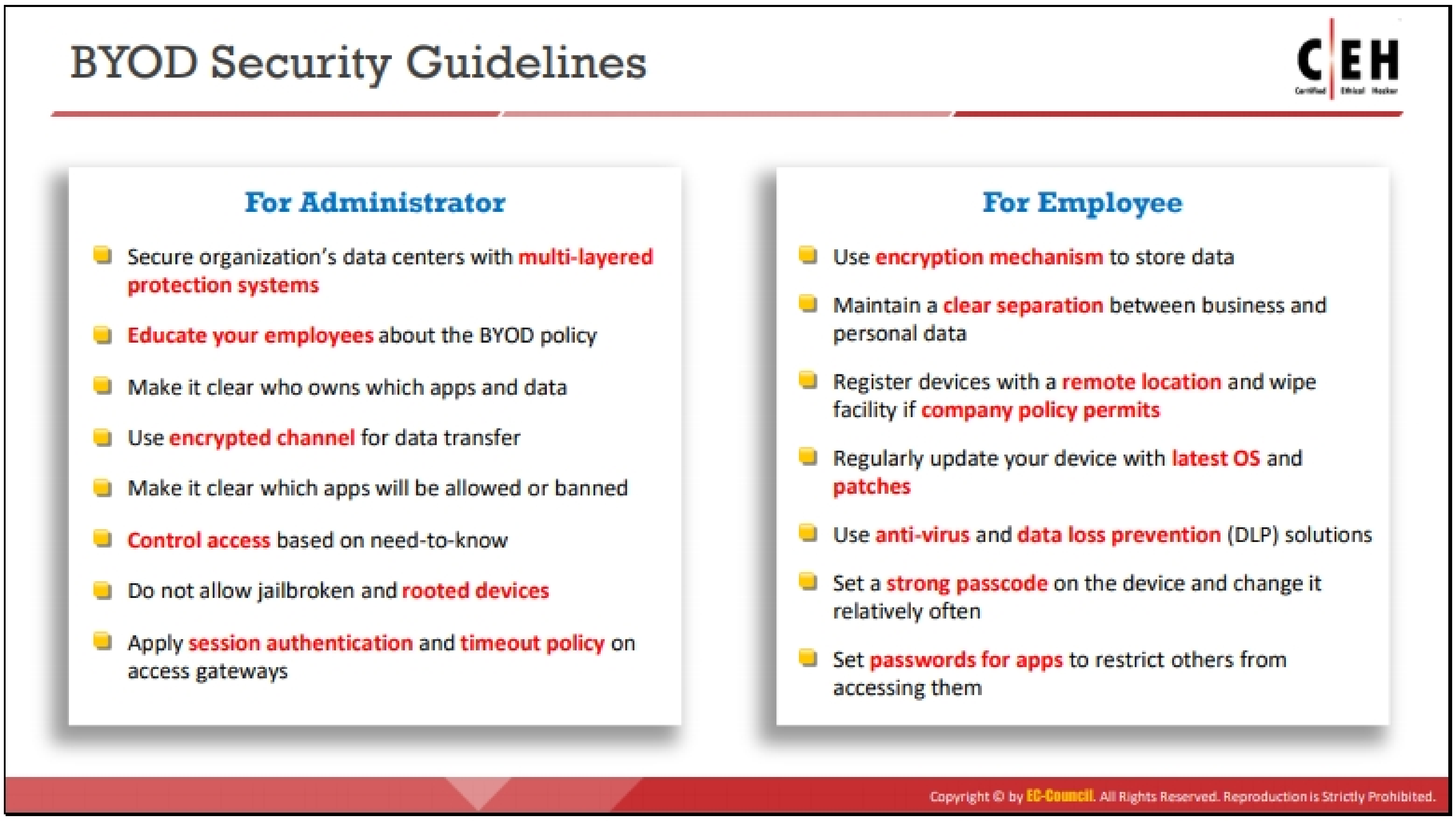
  Mobile management technology is effective only when policies are established, implemented, and supported. It is essential for organizations to keep the mobile ecosystem adequately secure to make the BYOD programs work. This requires a thorough assessment of the operating environment and the development of a solution that provides for the following: asset and identity management, local storage controls, removable media controls, network access levels, network application controls, corporate versus personal app controls, web and messaging security, device health management, data loss prevention, and so on.

  Mainly consider assessing and documenting risks in the following aspects:

  - Information security (for data, application, and user segment)
  - Operations security (for protecting user information)
  - Transmission security (for secure data transmission)

- **Support**

  The inconsistent nature of BYOD users will increase the frequency of support calls. Organizations should establish the process and capabilities in the early stages to ensure success. Mobile committees should frequently reassess the support levels and ensure the productivity of their mobile employees.



BYOD Security Guidelines

For Administrator

- Secure organization's data centers with **multi-layered protection systems**
- **Educate your employees** about the BYOD policy
- Make it clear who owns which apps and data
- Use **encrypted channel** for data transfer
- Make it clear which apps will be allowed or banned
- **Control access** based on need-to-know
- Do not allow jailbroken and **rooted devices**
- Apply **session authentication** and **timeout policy** on access gateways

For Employee

- Use **encryption mechanism** to store data
- Maintain a **clear separation** between business and personal data
- Register devices with a **remote location** and wipe facility if **company policy permits**
- Regularly update your device with **latest OS** and **patches**
- Use **anti-virus** and **data loss prevention** (DLP) solutions
- Set a **strong passcode** on the device and change it relatively often
- Set **passwords for apps** to restrict others from accessing them

BYOD Security Guidelines

- **For Administrator**

  With the increasing use of tablets, smartphones, and other devices at work, mobile security has emerged as a major concern. Listed below are security guidelines that an administrator should follow to secure the organization's network and data:

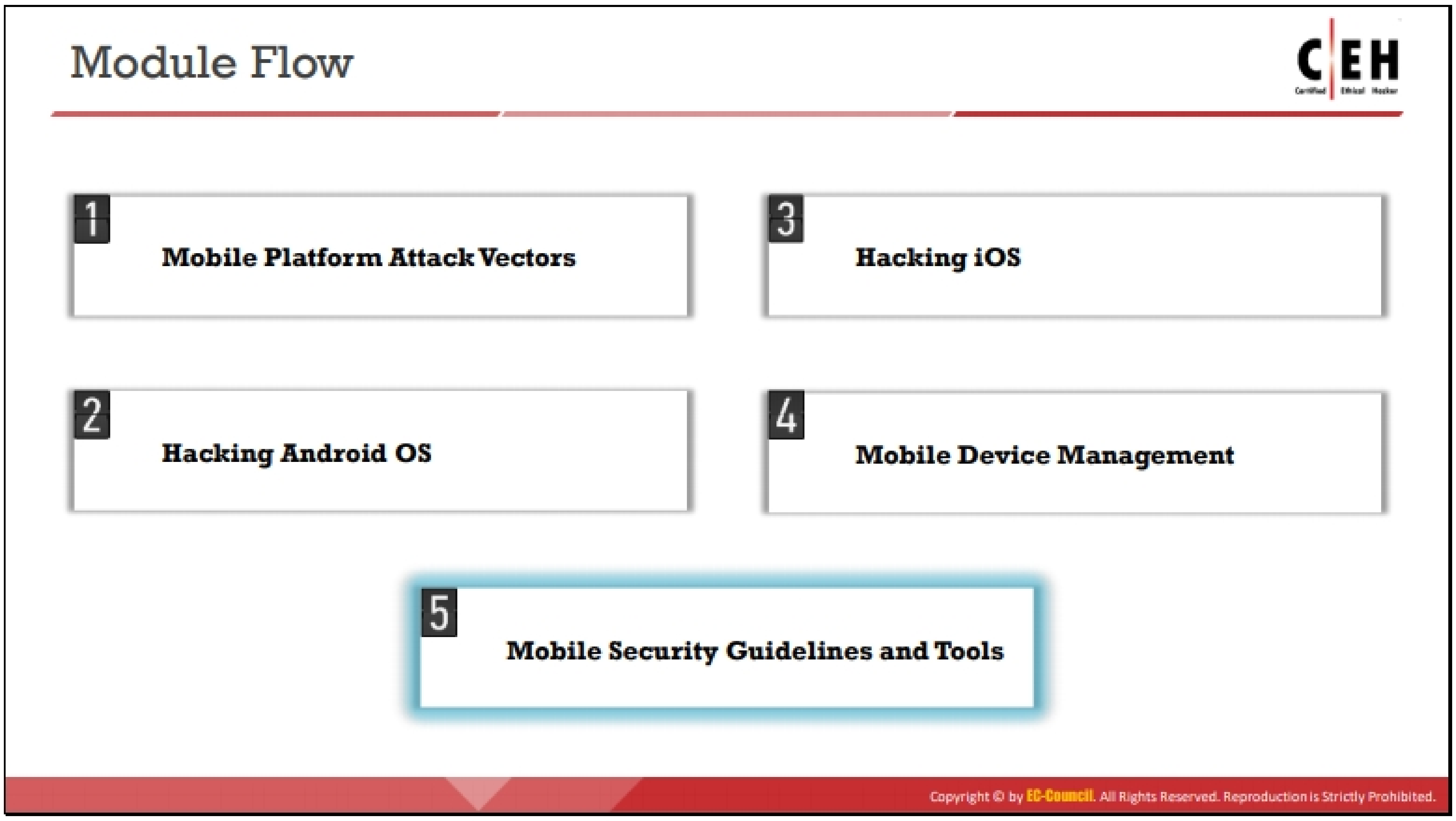
  - Secure the organization's data centers with multi-layered protection systems
  - Educate employees about the BYOD policy
  - Make it clear who owns what apps and data
  - Use encrypted channel for data transfer
  - Make it clear what apps will be allowed or banned
  - Control access based on a need-to-know basis
  - Do not allow jailbroken and rooted devices
  - Apply session authentication and timeout policy to access gateways
  - Impose company WLAN access when on site
  - Make users use complex passcodes and change them often
  - Ensure that the user's mobile device is registered and authenticated before allowing access to the organization's network
  - Consider multi-factor authentication methods to enhance security when remotely accessing the organization's information systems

  - Make users agree to and sign the BYOD policy before they can access the organization's information system
  - When an employee leaves the organization, state whether total device wipe or selective wipe of certain apps and data are required. In addition, ensure that the organization's data are maintained separately from the user's personal data.
  - Implement strong algorithms to encrypt all the organization's data stored in the user's mobile device; use an encrypted channel for data transfer.
  - In case the user's mobile device is lost or stolen, remotely reset or wipe the device passwords to prevent unauthorized access to the organization's sensitive data.
  - Implement an SSL-based VPN, which provides secure remote access
  - Ensure that users' devices are regularly updated with the latest OS and other software, which could avoid and sometimes even fix any security vulnerabilities.
  - Do not provide offline access to the organization's sensitive information, which should be accessible only via the company's network.
  - Enable a periodic re-authentication mechanism to ensure that a legitimate user is accessing the device.

- **For Employee**

  Listed below are the guidelines that an employee should follow to secure sensitive personal or corporate information stored on a mobile device:

  - Use encryption mechanisms to store data
  - Maintain a clear separation between business and personal data
  - Register devices with a remote locate and wipe facility if the company policy permits
  - Regularly update one's device with latest OS and patches
  - Use anti-virus and data loss prevention (DLP) solutions
  - Set a strong passcode for the device and change it often
  - Use strong algorithms to encrypt data
  - Set passwords for apps to restrict others from accessing them
  - Do not download files from untrusted sources
  - Be cautious while browsing websites and opening links or attachments sent via email



Module Flow

1 Mobile Platform Attack Vectors

2 Hacking Android OS

3 Hacking iOS

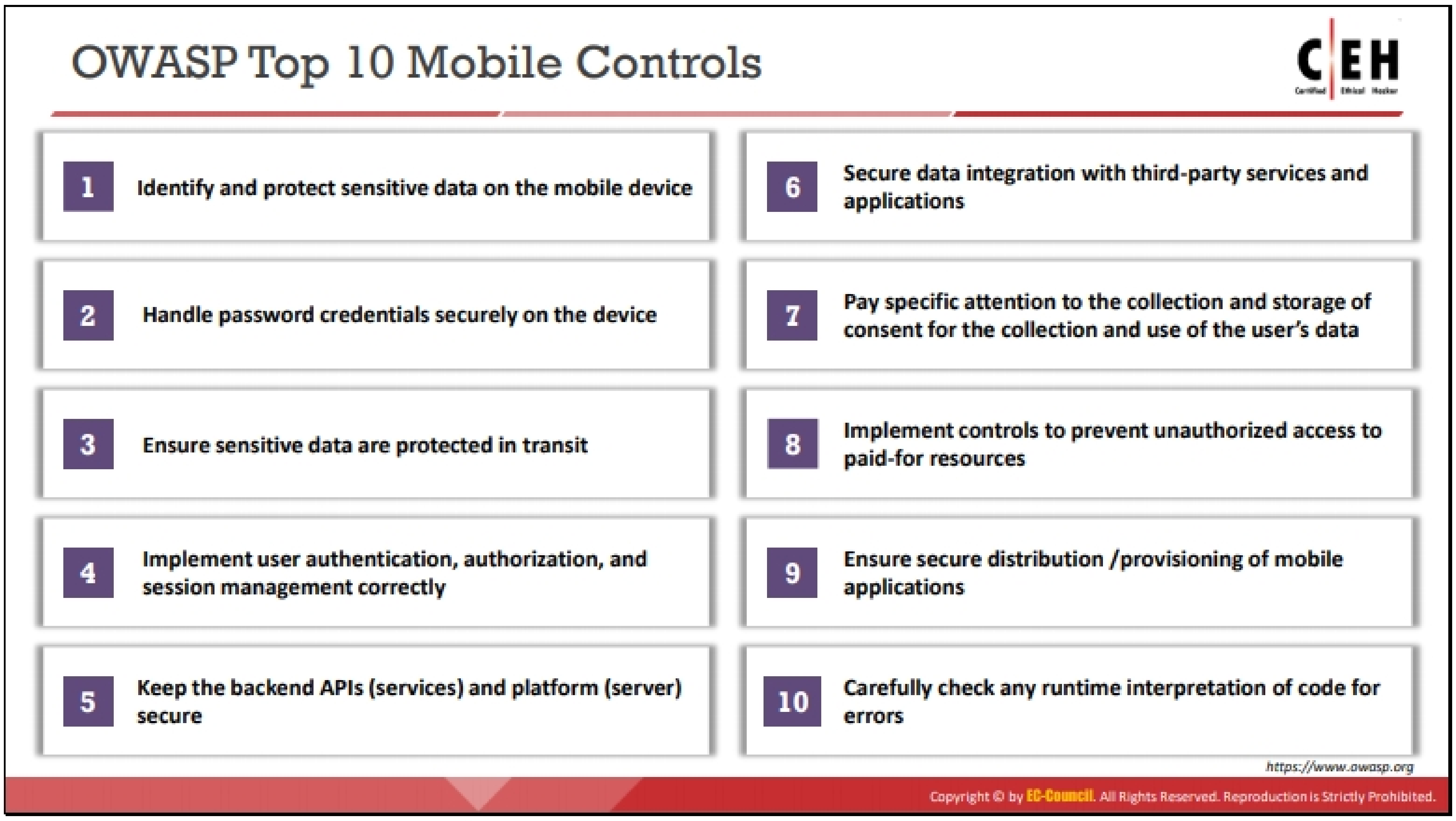
4 Mobile Device Management

5 Mobile Security Guidelines and Tools

Mobile Security Guidelines and Tools

Like personal computers, mobile devices store sensitive data and may be susceptible to various threats. Therefore, it is best to secure them to prevent the compromise or loss of confidential data, to reduce the risk of various threats such as viruses and Trojans, and to mitigate other forms of abuse. To secure these devices, one should adopt strict measures and use security tools.

This section deals with various mobile security guidelines and mobile protection tools that help to secure mobile devices.



OWASP Top 10 Mobile Controls

1. Identify and protect sensitive data on the mobile device
2. Handle password credentials securely on the device
3. Ensure sensitive data are protected in transit
4. Implement user authentication, authorization, and session management correctly
5. Keep the backend APIs (services) and platform (server) secure
6. Secure data integration with third-party services and applications
7. Pay specific attention to the collection and storage of consent for the collection and use of the user's data
8. Implement controls to prevent unauthorized access to paid-for resources
9. Ensure secure distribution /provisioning of mobile applications
10. Carefully check any runtime interpretation of code for errors

https://www.owasp.org

Mobile Security Guidelines

OWASP Top 10 Mobile Controls

Source: *https://www.owasp.org*

1. **Identify and protect sensitive data on the mobile device**
   - In the design phase, classify the data storage according to the sensitivity and then apply the controls. Process, store, and use data based on its classification
   - Apply validation of the security of API calls to the sensitive data
   - Store the sensitive data on the server instead of the client-side device, as it supports secure network connectivity and other protection mechanisms
   - Use file encryption API provided by the OS or other trusted source when storing data in a device.
   - Use encryption to store sensitive data and store it in a tamper-proof area if possible
   - Restrict access to sensitive data based on contextual information, e.g., location
   - Always make sure to turn off the location, GPS tracking, or other sensitive information when not in use
   - Always be aware of the public shared storage as it is easily vulnerable to data leakage
   - Apply the principle of minimal disclosure and identify the type of data needed in the design phase

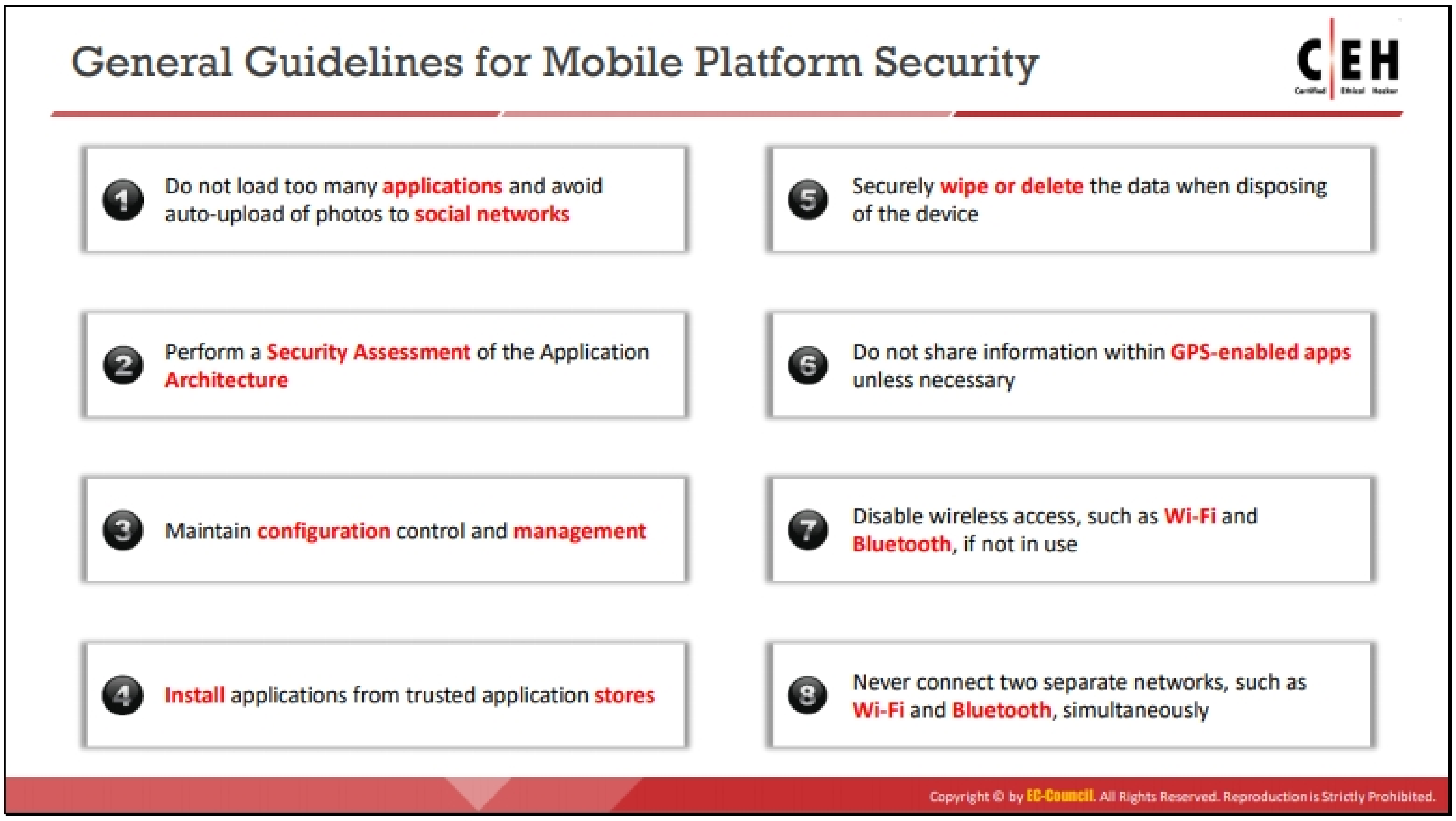
- Use non-persistent identifiers wherever possible, which are not shared with other apps
- Applications should use remote wipe and kill switch APIs for removing sensitive information from the device in the event of theft or loss

2. **Handle password credentials securely on the device**
   - Use longer term authorization tokens instead of passwords as per the OAuth model and encrypt tokens in transit using SSL/TLS
   - Leverage the encryption and key-store mechanisms provided by the mobile OS to securely store passwords and authorization tokens
   - Ensure that capabilities such as secure element are used to store keys, credentials, and other sensitive data
   - Allow access to mobile users for changing the passwords on the device
   - Make sure to use measures that allow repeated patterns to curb smudge attacks
   - Make sure that no password or key is visible in the cache or logs
   - Do not store any passwords or secrets in the mobile application binaries, as they can be easily downloaded and reverse engineered
3. **Ensure sensitive data are protected in transit**
   - Enforce the use of an end-to-end secure channel such as SSL/TLS when sending sensitive information over the network
   - Use complex and well-known encryption algorithms such as AES with appropriate key lengths for enhanced security
   - Ensure the use of certificates signed by trusted CA providers and do not disable or ignore SSL chain validation
   - A secure connection should be established only after verifying the identity of the remote end point for reducing the risk of MITM attacks
   - Sending sensitive data using SMS or MMS from or to the mobile end points should be avoided
4. **Implement user authentication, authorization, and session management correctly**
   - The authentication mechanism strength must depend on the sensitivity of the data being processed by the application and its access to valuable resources
   - Ensure that session management is handled properly after the initial authentication using appropriate secure protocols
   - Use unpredictable session identifiers with high entropy and repeated application of SHA1 for combining random variables
   - Use contexts such as IP location for adding security to authentication

- Ensure the use of additional authentication factors for mobile applications that give access to sensitive data using voice, fingerprint, or other behavioral inputs
- Use authentication that depends on the end-user identity rather than the device identity

5. **Keep the backend APIs (services) and the platform (server) secure**
    - Perform detailed code checking for sensitive data that is transferred unintentionally between the mobile device, web-server backend, and other external interfaces
    - All the backend services for the mobile apps should be tested for vulnerabilities periodically using any static code analyzer tools and fuzzing tools
    - Ensure that the backend platform is running with a hardened configuration with the latest security patches applied to the OS and web server
    - Adequate logs are reserved at the backend for detecting and responding to incidents and for performing forensics
    - Use rate limiting and throttling on a per-user/IP basis for reducing the risk of DDoS attacks
    - Ensure testing for DoS vulnerabilities that make the server flooded with resource-intensive application calls
    - Perform use case testing and abuse case testing to determine the vulnerabilities; also perform testing of the backend web services/REST
6. **Secure data integration with third-party services and applications**
    - Always scrutinize the authenticity of any third-party code or libraries used in the mobile application
    - Regularly update the latest security patches and keep track of all the third-party APIs and framework
    - Validate all the data received and sent before processing for non-trusted third-party applications
7. **Pay specific attention to the collection and storage of consent for the collection and use of the user's data**
    - Create a privacy policy that covers the usage of personal data and make it available to users when making consent choices such as at install time or at run time or via opt-out mechanisms
    - Check if any application is collecting Personally Identifiable Information (PII)
    - Review the communication mechanisms to check for any accidental leaks
    - Always preserve the record of consent to the transfer of PII
    - Ensure that the consent collection mechanism does not overlap or conflict and try to resolve any conflicts

8. **Implement controls to prevent unauthorized access to paid-for resources (wallet, SMS, phone calls, etc.)**
    - Maintain access logs to paid-for resources in a non-repudiable format and make them available for end-user monitoring
    - Regularly check for any abnormal usage patterns in paid-for resource usage and activate re-authentication
    - Ensure use of the white-list model by default for addressing paid-for resources
    - Authenticate all the API calls to paid-for resources
    - Ensure that the wallet API callbacks do not permit cleartext passwords and other sensitive information
    - Caution users and obtain permission for any type of cost implications for app performance
    - Implement best practices such as low latency and caching to minimize the signaling load on the base stations
9. **Ensure secure distribution/provisioning of mobile applications**
    - The applications must be designed and provisioned to allow updates for security patches
    - The app stores should monitor the apps for vulnerable code and should be able to remove apps remotely at short notice in the case of an incident
    - Provide a feedback channel for the users to report security problems with the apps
10. **Carefully check any runtime interpretation of code for errors**
    - Minimize runtime interpretation and the capabilities offered to runtime interpreters and run interpreters with minimum privileges
    - Outline comprehensive escape syntax as appropriate
    - Use fuzz test interpreters and sandbox interpreters



General Guidelines for Mobile Platform Security

① Do not load too many applications and avoid auto-upload of photos to social networks

② Perform a Security Assessment of the Application Architecture

③ Maintain configuration control and management

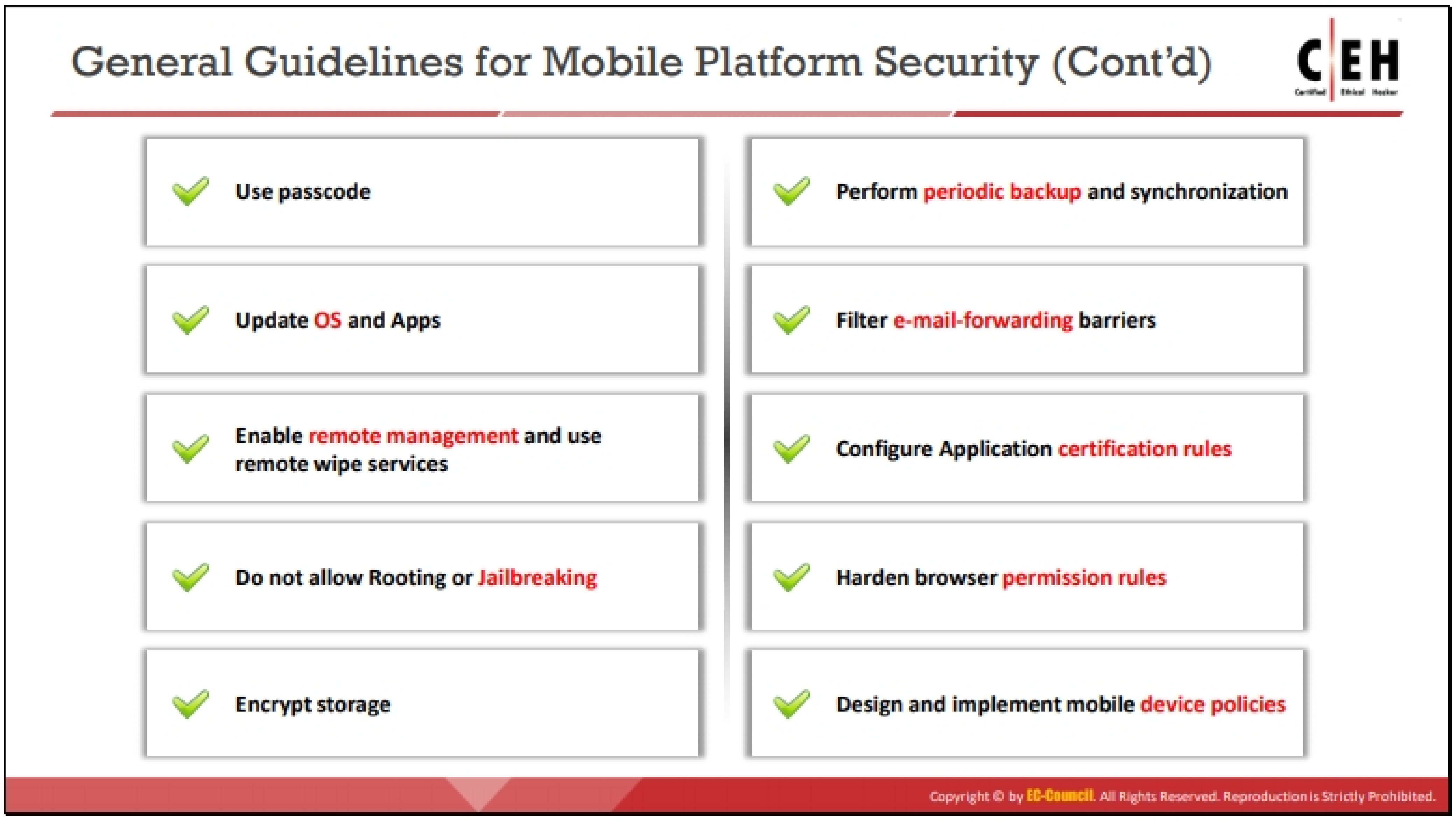
④ Install applications from trusted application stores

⑤ Securely wipe or delete the data when disposing of the device

⑥ Do not share information within GPS-enabled apps unless necessary

⑦ Disable wireless access, such as Wi-Fi and Bluetooth, if not in use

⑧ Never connect two separate networks, such as Wi-Fi and Bluetooth, simultaneously



General Guidelines for Mobile Platform Security (Cont'd)

- ✓ **Use passcode**
- ✓ **Update OS and Apps**
- ✓ **Enable remote management and use remote wipe services**
- ✓ **Do not allow Rooting or Jailbreaking**
- ✓ **Encrypt storage**
- ✓ **Perform periodic backup and synchronization**
- ✓ **Filter e-mail-forwarding barriers**
- ✓ **Configure Application certification rules**
- ✓ **Harden browser permission rules**
- ✓ **Design and implement mobile device policies**

General Guidelines for Mobile Platform Security

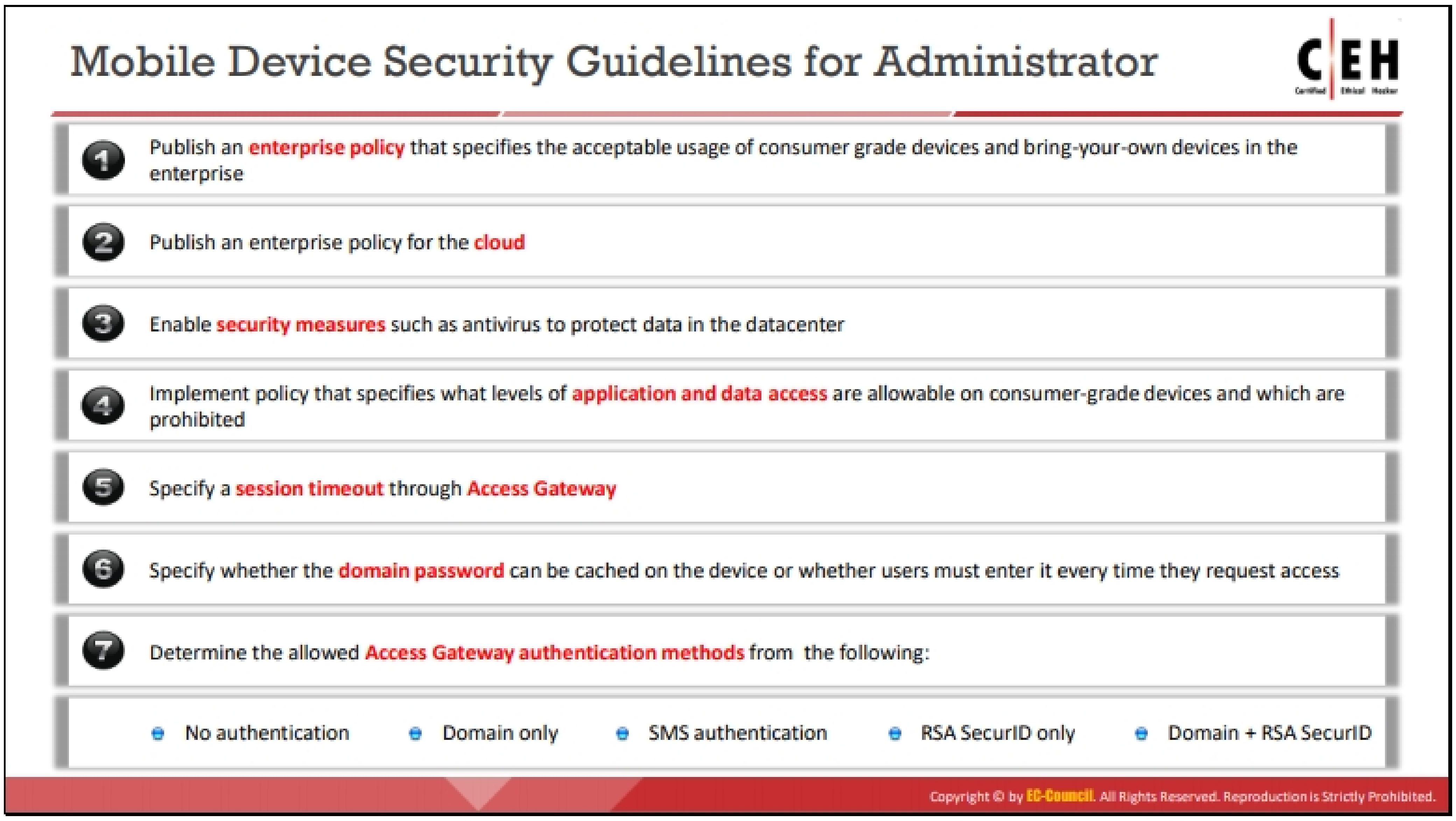
Given below are various guidelines that help you to protect your mobile device:

- Do not load too many applications and avoid auto-upload of photos to social networks
- Perform a security assessment of the application architecture
- Maintain configuration control and management

- Install applications from trusted application stores
- Securely wipe or delete the data when disposing of the device
- Do not share the information within GPS-enabled apps unless it is necessary
- Never connect two separate networks such as Wi-Fi and Bluetooth simultaneously
- Disable wireless access such as Wi-Fi and Bluetooth if not in use
    - Ensure that your Bluetooth is "off" by default. Turn it on whenever it is necessary
    - Disable wireless access such as Wi-Fi and Bluetooth if not in use to avoid illegal wireless access to the device
    - Disable sharing/tethering Internet connections over Wi-Fi and Bluetooth when not in use
- **Use Passcode**
    - Configure a strong passcode with the maximum possible length to gain access to your mobile devices
    - Set an idle timeout to automatically lock the phone when not in use
    - Enable the lockout/wipe feature after a certain number of attempts
    - Consider an eight-character complex passcode
    - Thwart passcode guessing: set erase data to ON
- **Update OS and Apps**
    - Update OS and apps to keep them secure
    - Apply software updates when new releases are available
    - Perform regular software maintenance
- **Enable Remote Management**
    - In an enterprise environment, use MDM software to secure, monitor, manage, and support mobile devices deployed across the organization
- **Do not allow Rooting or Jailbreaking**
    - Ensure that your MDM solutions prevent or detect rooting/jailbreaking
    - Include this clause in your mobile security policy
- **Use Remote Wipe Services**
    - Use remote wipe services such as Find My Device (Android) and Find My iPhone or FindMyPhone (Apple iOS) to locate your device should it be lost or stolen
    - Report a lost or stolen device to IT so that they can disable certificates and other access methods associated with the device

- **Encrypt Storage**
    - If supported, configure your mobile device to encrypt its storage with hardware encryption
    - Use device encryption and patch applications
    - Encrypt the device and backups
- **Perform periodic backup and synchronization**
    - Use a secure, over-the-air backup-and-restore tool that performs periodic background synchronization
    - (Android) Backup to your Google account so that sensitive enterprise data are not backed up to the cloud
    - Control the location of backups
    - Encrypt backups
    - Keep sensitive data off shared mobile devices. If enterprise information is locally stored on a device, then it is recommended that this device not be openly shared
    - Limit logging data stored on the device
    - Use a secure data-transfer utility or encrypt data in transit to or from the device, to ensure confidentiality and data integrity
- **Filter email-forwarding barriers**
    - Filter emails by configuring server-side settings of the corporate email system
    - Use commercial data loss prevention filters
    - Prevent local caching of email
- **Configure Application certification rules**
    - Allow only signed applications to install or execute
    - Configure wireless to ask to join networks
    - Sandbox applications and data
    - Enable auto-lock and set the timeout to one minute
    - Consider the privacy implications before enabling location-based services and limit usage to trusted applications
    - Configure location services to disable location tracking for applications that you do not want to know your location information
    - Configure notifications to disable the ability to view notifications while the device is locked for applications that could display sensitive data

  - Configure Auto Fill: Auto-fill names and passwords for browsers to reduce password loss via shoulder-surfing and surveillance (if desired and allowed by the enterprise policy)
  - Disable the collection of diagnostics and usage data under **Settings → General → About**
- **Harden browser permission rules**
  - Harden browser permission rules according to the company's security policies to avoid attacks
- **Design and implement mobile device policies**
  - Set a policy that defines the accepted usage, levels of support, and type of information access permitted on different devices
- Control devices and applications
- Prohibit USB keys
- Manage operating and application environments
- Press the power button to lock the device whenever it is not in use
- Verify the location of printers before printing sensitive documents
- Ask your IT department how to use Citrix technologies to keep data in the data center and personal devices personal
- If you must have sensitive data on a mobile device, use follow-me data and ShareFile as an enterprise-managed solution



Mobile Device Security Guidelines for Administrator

1. Publish an **enterprise policy** that specifies the acceptable usage of consumer grade devices and bring-your-own devices in the enterprise
2. Publish an enterprise policy for the **cloud**
3. Enable **security measures** such as antivirus to protect data in the datacenter
4. Implement policy that specifies what levels of **application and data access** are allowable on consumer-grade devices and which are prohibited
5. Specify a **session timeout** through **Access Gateway**
6. Specify whether the **domain password** can be cached on the device or whether users must enter it every time they request access
7. Determine the allowed **Access Gateway authentication methods** from the following:

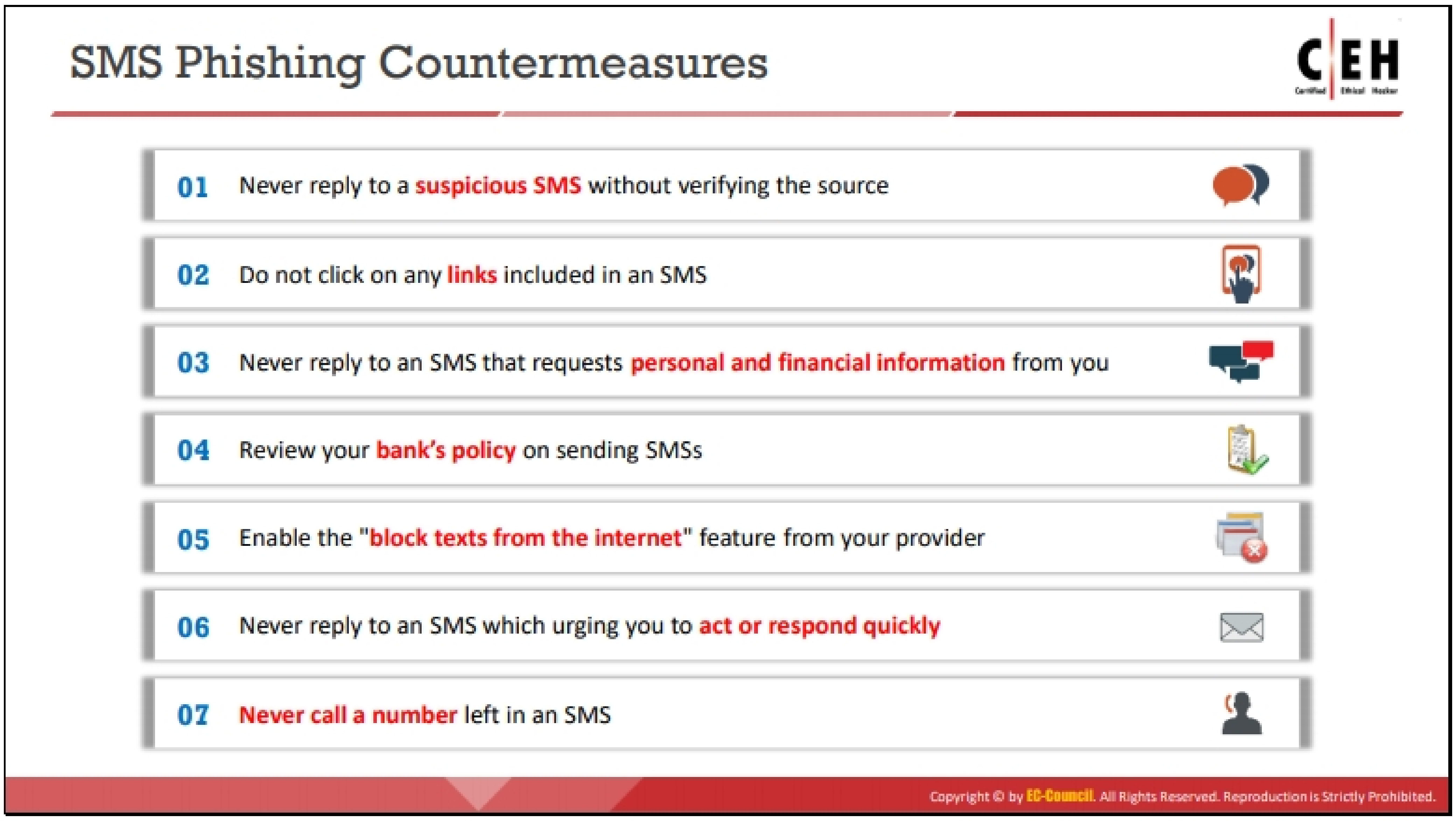
- No authentication
- Domain only
- SMS authentication
- RSA SecurID only
- Domain + RSA SecurID

Mobile Device Security Guidelines for Administrator

Listed below are some guidelines that an administrator can implement to maintain corporate mobile device security:

- Publish an enterprise policy that specifies the acceptable usage of consumer-grade devices and BYOD in the enterprise
- Publish an enterprise policy for the cloud
- Enable security measures such as anti-virus to protect the data in the data center
- Implement a policy that specifies what levels of application and data access are allowable on consumer-grade devices and which ones are prohibited
- Specify a session timeout through Access Gateway
- Specify whether the domain password can be cached on the device or whether users must enter it every time they request access
- Determine the allowed Access Gateway authentication methods from the following:
  - No authentication
  - Domain only
  - SMS authentication
  - RSA SecurID only
  - Domain + RSA SecurID

- Develop and maintain a mobile device security policy that states organizational resources to be accessed via mobile devices, types of mobile devices allowed, access privileges, and others
- Develop system threat models for mobile devices and the resources accessed using them, which enable an organization to design security solutions
- Enable all the required security settings for mobile devices prior to issuing them to users
- Regularly maintain mobile device security, including keeping the OS and apps up to date, ensuring that mobile clocks are synched to a common time source, reconfiguring access privileges, identifying and documenting abnormalities within device infrastructures, etc.
- Regularly monitor whether users properly follow policies and procedures framed for device security
- Consider the best services provided by various service providers, determine the services that suit your environment, and then design and attain one or more solutions to meet these and any other requirements
- Test the solutions prior to placing them into production. Evaluate various aspects of solutions such as authentication, app functionality, security, connectivity, and performance



SMS Phishing Countermeasures

01 Never reply to a **suspicious SMS** without verifying the source

02 Do not click on any **links** included in an SMS

03 Never reply to an SMS that requests **personal and financial information** from you

04 Review your **bank's policy** on sending SMSs

05 Enable the "**block texts from the internet**" feature from your provider

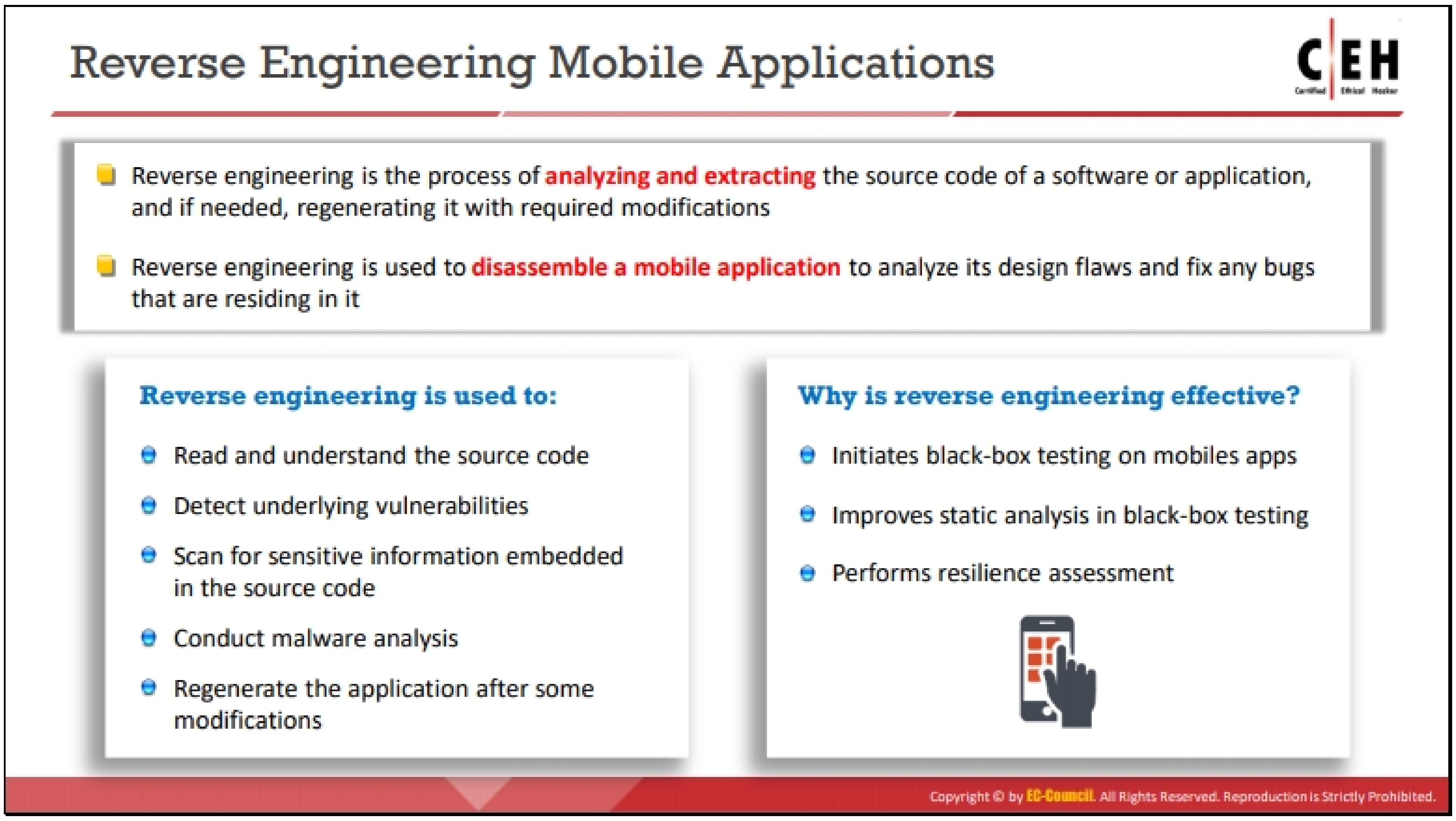
06 Never reply to an SMS which urging you to **act or respond quickly**

07 **Never call a number** left in an SMS

SMS Phishing Countermeasures

Countermeasures for defending against SMS phishing attacks are listed below:

- Never reply to a suspicious SMS without verifying the source
- Do not click on any links included in the SMS
- Never reply to an SMS that requires personal and financial information from you
- Review the bank's policy on sending SMS
- Enable the "block texts from the internet" feature from your provider
- Never reply to an SMS that urges you to act or respond quickly
- Never call a number left in an SMS
- Do not fall for scams, gifts, and offers that seem to be unexpected
- Attackers might send text messages through an Internet text relay service to conceal their identity; thus, it is best to avoid messages from nontelephonic numbers
- Check for spelling mistakes, grammatical errors, or language inconsistency in text messages



Reverse Engineering Mobile Applications

- Reverse engineering is the process of **analyzing and extracting** the source code of a software or application, and if needed, regenerating it with required modifications
- Reverse engineering is used to **disassemble a mobile application** to analyze its design flaws and fix any bugs that are residing in it

Reverse engineering is used to:

- Read and understand the source code
- Detect underlying vulnerabilities
- Scan for sensitive information embedded in the source code
- Conduct malware analysis
- Regenerate the application after some modifications

Why is reverse engineering effective?

- Initiates black-box testing on mobiles apps
- Improves static analysis in black-box testing
- Performs resilience assessment

Reverse Engineering Mobile Applications

Reverse engineering is the process of analyzing and extracting the source code of a software or application and, if needed, regenerating it with required modifications. Reverse engineering is used to disassemble a software program or a mobile application to analyze its design flaws and fix any bugs that are residing in it. The technique is also used for discovering underlying vulnerabilities and improving defense strategies against attacks. Reverse engineering can also be used in mobile platforms to create duplicate or clone apps.

Reverse engineering is used to:

- Read and understand the source code
- Detect underlying vulnerabilities
- Scan for sensitive information embedded inside the source code
- Conduct malware analysis
- Regenerate the application or software after applying some modifications

Why reverse engineering is effective?

Mobile security professionals need to possess basic knowledge of reverse engineering techniques for the below reasons:

- **Initiating black-box testing on mobiles apps**

  Current mobile apps incorporate controls that do not allow dynamic analysis. End-to-end encryption and SSL lead to obstacles in intercepting and modifications. Root detection restricts apps from operating on a rooted device. It can also hinder the usage of advanced tools for testing. These defenses should be neutralized to analyze the source code.

- **Improving static analysis in black-box testing**

  In black-box testing, the underlying design and operation of the application can be comprehended by static analysis of the binary code and bytecode of the app. The process can also help in discovering vulnerabilities in hardcoded credentials.

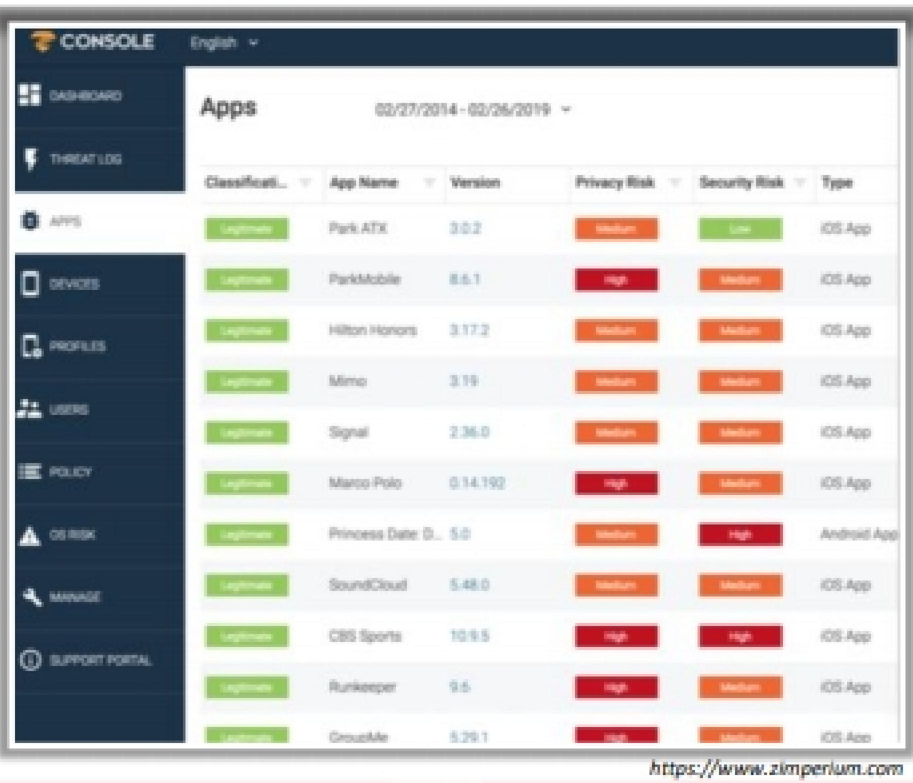
- **Performing resilience assessment**

  Apps must be designed to withstand reverse engineering by implementing software protection methods such as Mobile Application Security Verification Standard Anti-Reversing Controls (MASVS-R). Efficiency of the controls can be verified by conducting general testing methods such as resilience assessment. Security professionals need to perform a resilience assessment by conducting reverse engineering and trying to breach the mobile application's security defenses.

Source Code Analysis Tools

z3A Advanced App Analysis

z3A Advanced App Analysis allows security professionals to **identify security and privacy risks** across various iOS and Android applications



Apps 02/27/2014 - 02/26/2019

Classificati...	App Name	Version	Privacy Risk	Security Risk	Type
Legitimate	Park ATX	3.0.2	Medium	Low	iOS App
Legitimate	ParkMobile	8.6.1	High	Medium	iOS App
Legitimate	Hilton Honors	3.17.2	Medium	Medium	iOS App
Legitimate	Mimo	3.19	Medium	Medium	iOS App
Legitimate	Signal	2.36.0	Medium	Medium	iOS App
Legitimate	Marco Polo	0.14.192	High	Medium	iOS App
Legitimate	Princess Date: D...	5.0	Medium	High	Android App
Legitimate	SoundCloud	5.48.0	Medium	Medium	iOS App
Legitimate	CBS Sports	10.9.5	High	High	iOS App
Legitimate	Runkeeper	9.6	High	Medium	iOS App
Legitimate	GroupMe	5.29.1	High	Medium	iOS App

https://www.zimperium.com

Kiuwan
https://www.kiuwan.com

Appium
http://appium.io

Selendroid
http://selendroid.io

Bitbar
https://bitbar.com

Infer
https://fbinfer.com

Mobile Security Tools

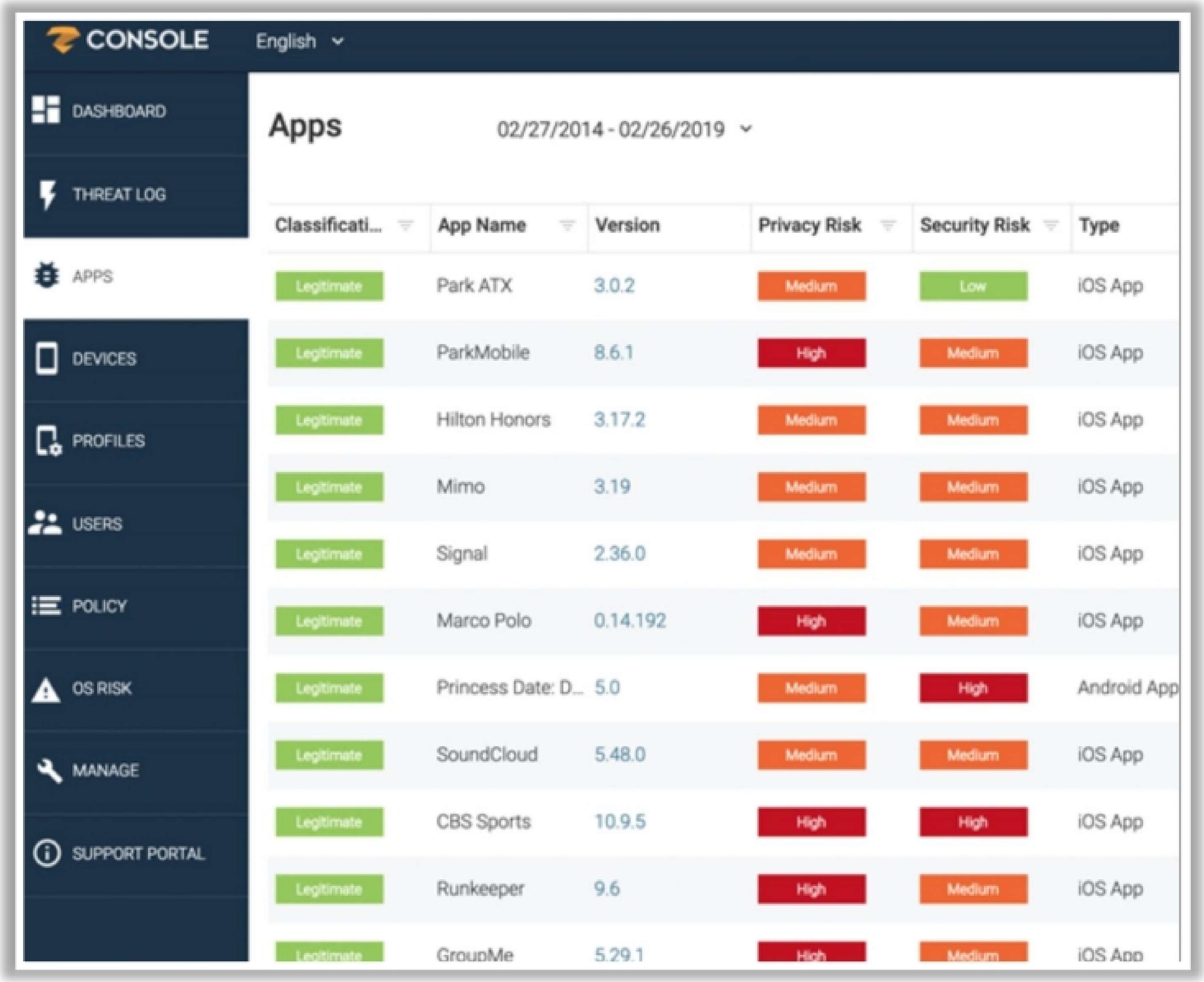
Unlike mobile devices of the past, today's mobiles come with advanced computing capability and connectivity (smartphones). One can use them to store data, browse the Internet, record videos, send SMS, play games, capture photos, and many other tasks. Therefore, mobile devices have become the major source for intruders to steal data. Various types of mobile security tools are discussed below.

Source Code Analysis Tools

- **z3A Advanced App Analysis**

  Source: *https://www.zimperium.com*

  z3A Advanced App Analysis allows security professionals to identify security and privacy risks across various iOS and Android applications. For each risky mobile app that is identified, Zimperium's z3ATM solution provides deep intelligence, including contextual analysis as well as privacy and security ratings. A parallel processing engine continuously collects and correlates data from multiple sources, i.e., from malware to data manipulation instances. Multivariate tests and validations are performed to identify mobile app security and privacy risks before they become threats.

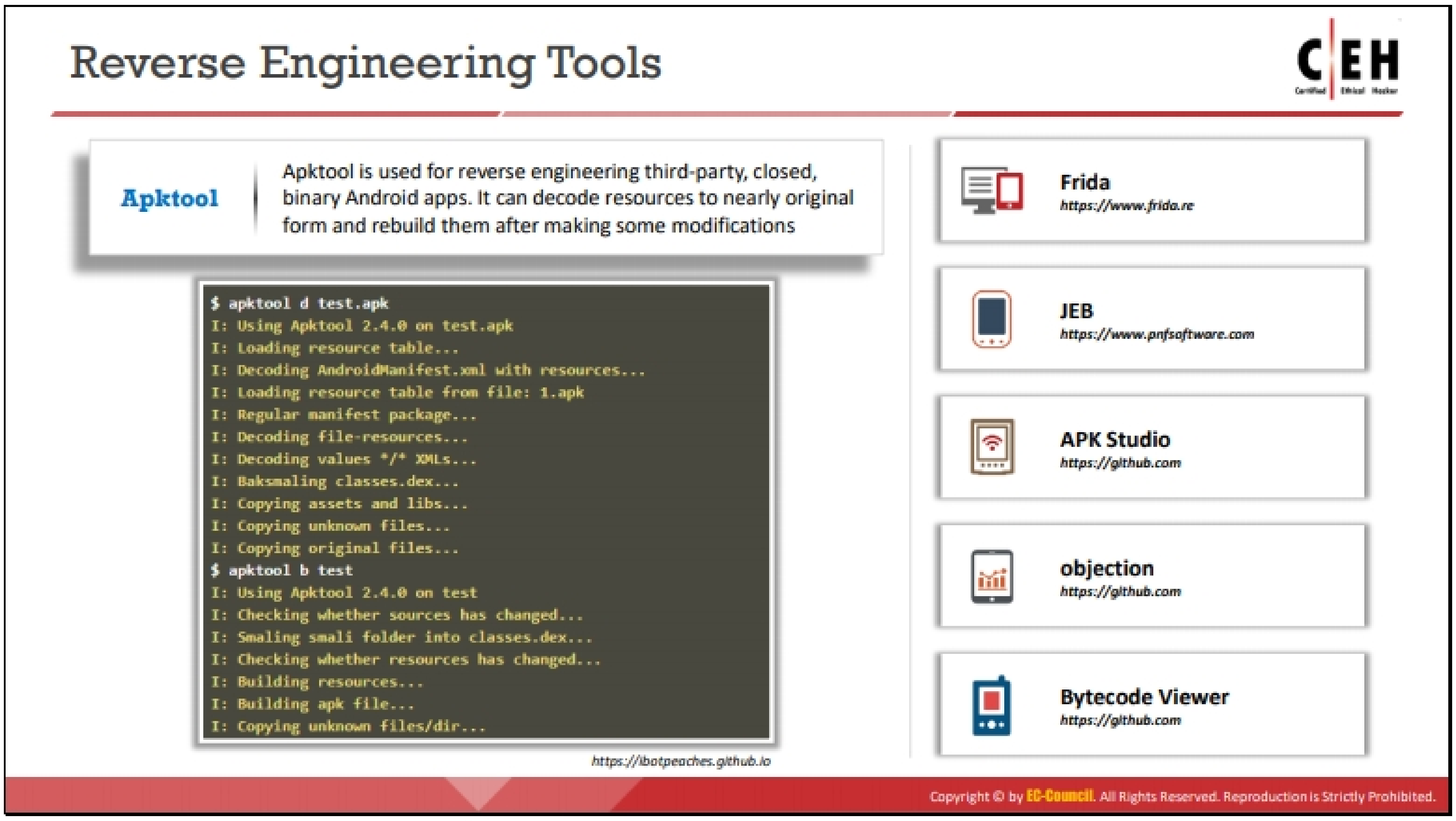

CONSOLE
English
DASHBOARD
THREAT LOG
APPS
DEVICES
PROFILES
USERS
POLICY
OS RISK
MANAGE
SUPPORT PORTAL
Apps
02/27/2014 - 02/26/2019

Classificati...	App Name	Version	Privacy Risk	Security Risk	Type
Legitimate	Park ATX	3.0.2	Medium	Low	iOS App
Legitimate	ParkMobile	8.6.1	High	Medium	iOS App
Legitimate	Hilton Honors	3.17.2	Medium	Medium	iOS App
Legitimate	Mimo	3.19	Medium	Medium	iOS App
Legitimate	Signal	2.36.0	Medium	Medium	iOS App
Legitimate	Marco Polo	0.14.192	High	Medium	iOS App
Legitimate	Princess Date: D...	5.0	Medium	High	Android App
Legitimate	SoundCloud	5.48.0	Medium	Medium	iOS App
Legitimate	CBS Sports	10.9.5	High	High	iOS App
Legitimate	Runkeeper	9.6	High	Medium	iOS App
Legitimate	GroupMe	5.29.1	High	Medium	iOS App

Figure 17.68: Screenshot of z3A Advanced App Analysis

Some additional tools used for source code analysis of mobile applications are listed below:

- Kiuwan (*https://www.kiuwan.com*)
- Appium (*http://appium.io*)
- Selendroid (*http://selendroid.io*)
- Bitbar (*https://bitbar.com*)
- Infer (*https://fbinfer.com*)



Reverse Engineering Tools

Apktool

Apktool is used for reverse engineering third-party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications

```
$ apktool d test.apk
I: Using Apktool 2.4.0 on test.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: 1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
$ apktool b test
I: Using Apktool 2.4.0 on test
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
```

https://ibotpeaches.github.io

Frida
https://www.frida.re

JEB
https://www.pnfsoftware.com

APK Studio
https://github.com

objection
https://github.com

Bytecode Viewer
https://github.com

Reverse Engineering Tools

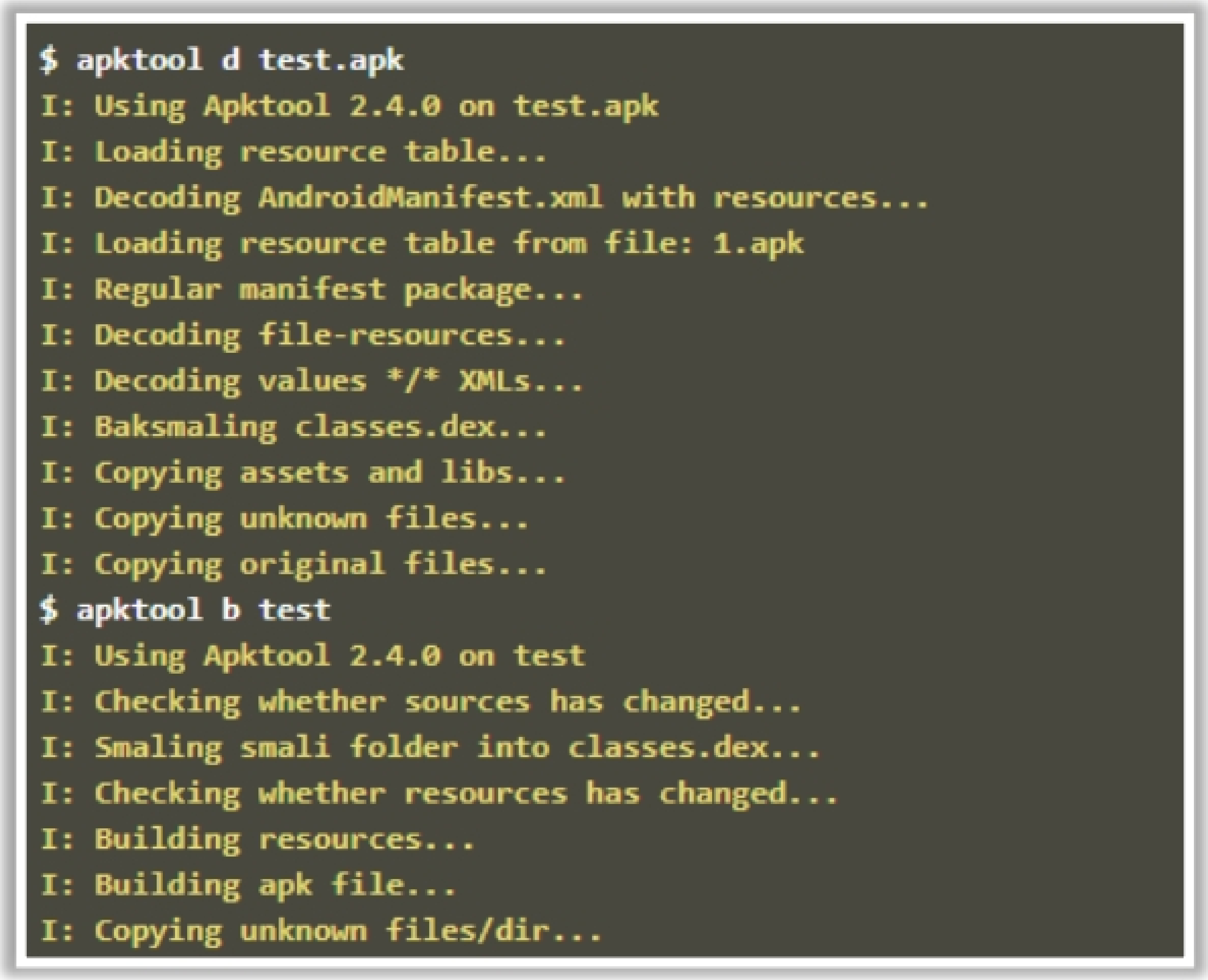
- **Apktool**

  Source: *http://www.javadecompilers.com*

  Apktool is used for reverse engineering third-party, closed, binary Android apps. It can decode resources nearly to their original form and rebuild them after making some modifications. It also makes working with an app easier because of the project-like file structure and automation of some repetitive tasks such as building APK, etc.

  Features:

  - Disassembling resources nearly to their original form
  - Rebuilding decoded resources back to binary APK/JAR
  - Organizing and handling APKs that depend on framework resources
  - Smali Debugging

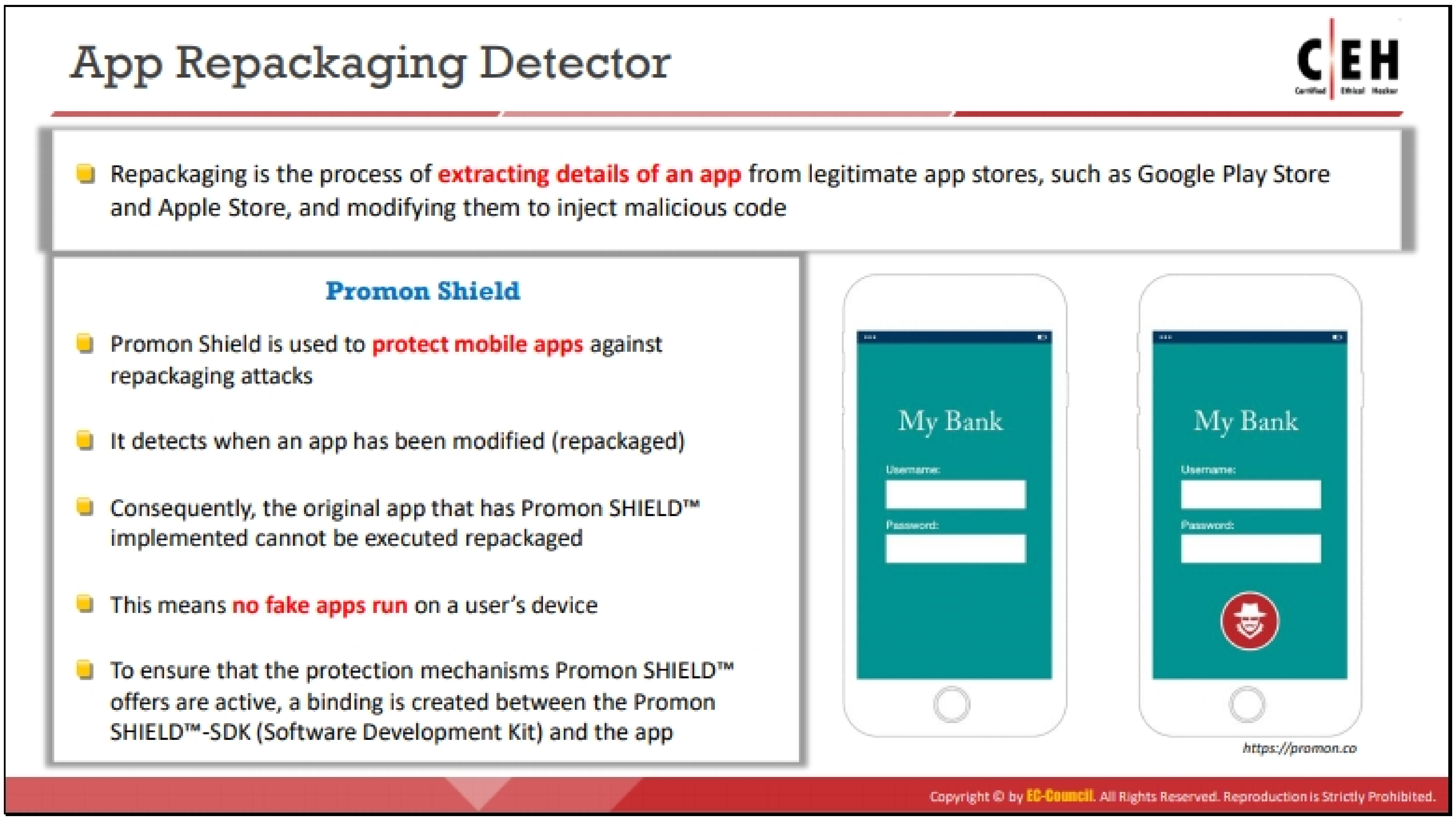


```
$ apktool d test.apk
I: Using Apktool 2.4.0 on test.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: 1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
$ apktool b test
I: Using Apktool 2.4.0 on test
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
```

Figure 17.69: Screenshot of Apktool

Some additional mobile application reverse engineering tools are listed below:

- Frida (*https://www.frida.re*)
- JEB (*https://www.pnfsoftware.com*)
- APK Studio (https://github.com)
- objection (*https://github.com*)
- Bytecode Viewer (*https://github.com*)



App Repackaging Detector

- Repackaging is the process of **extracting details of an app** from legitimate app stores, such as Google Play Store and Apple Store, and modifying them to inject malicious code

Promon Shield

- Promon Shield is used to **protect mobile apps** against repackaging attacks
- It detects when an app has been modified (repackaged)
- Consequently, the original app that has Promon SHIELD™ implemented cannot be executed repackaged
- This means **no fake apps run** on a user's device
- To ensure that the protection mechanisms Promon SHIELD™ offers are active, a binding is created between the Promon SHIELD™-SDK (Software Development Kit) and the app

https://promon.co

App Repackaging Detector

Repackaging is a process of extracting the details of an app from legitimate app stores such as Google Play Store and Apple Store and modifying the app by injecting malicious code. Then, the app is redistributed for public use as an authentic application. Repackaging can also be done during reverse engineering of an application.

- **Promon Shield**

  Source: *https://promon.co*

  Promon SHIELD™ is used to protect mobile apps against repackaging attacks. It detects when an app has been modified (repackaged). Consequently, the original app that has Promon SHIELD™ implemented cannot be executed when repackaged. This means that no fake apps can run on a user's device. To ensure that the protection mechanisms that Promon SHIELD™ offers are active, a binding between the Promon SHIELD™ software development kit (SDK) and the app is created.

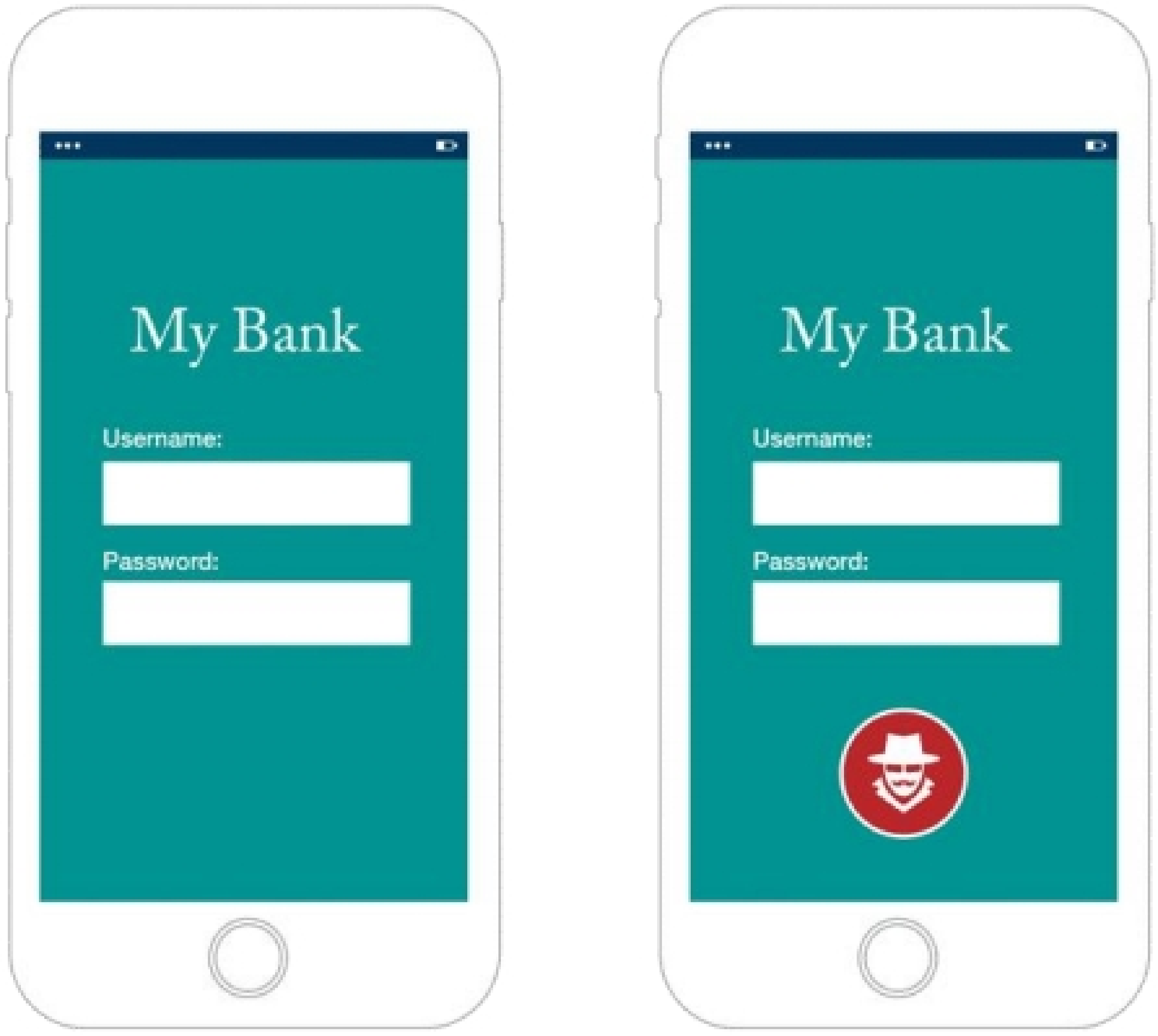
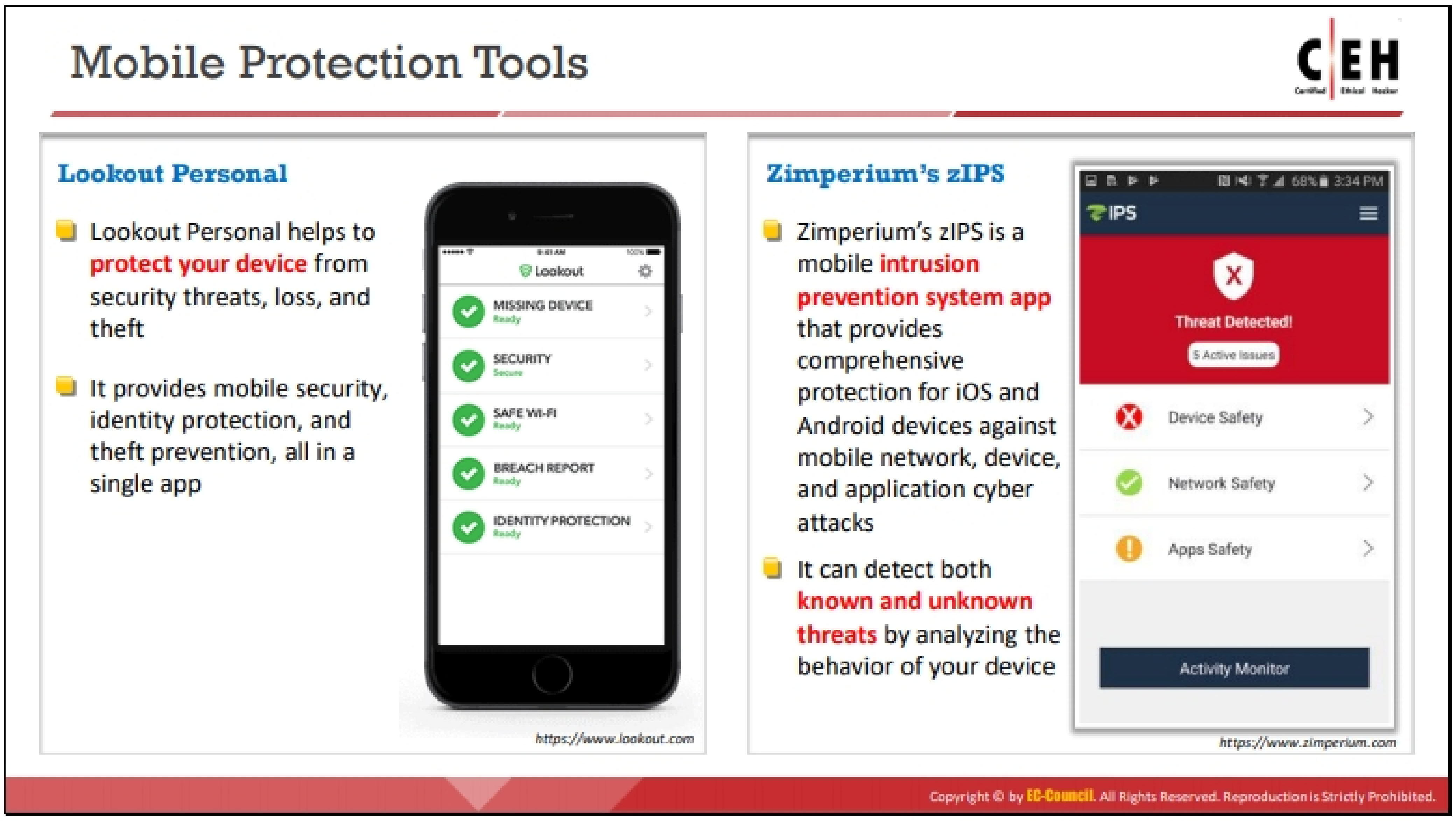

My Bank
Username:
Password:
My Bank
Username:
Password:

Figure 17.70: Screenshot of Promon Shield



Mobile Protection Tools

Lookout Personal

- Lookout Personal helps to protect your device from security threats, loss, and theft
- It provides mobile security, identity protection, and theft prevention, all in a single app

Lookout
MISSING DEVICE
Ready
SECURITY
Secure
SAFE WI-FI
Ready
BREACH REPORT
Ready
IDENTITY PROTECTION
Ready

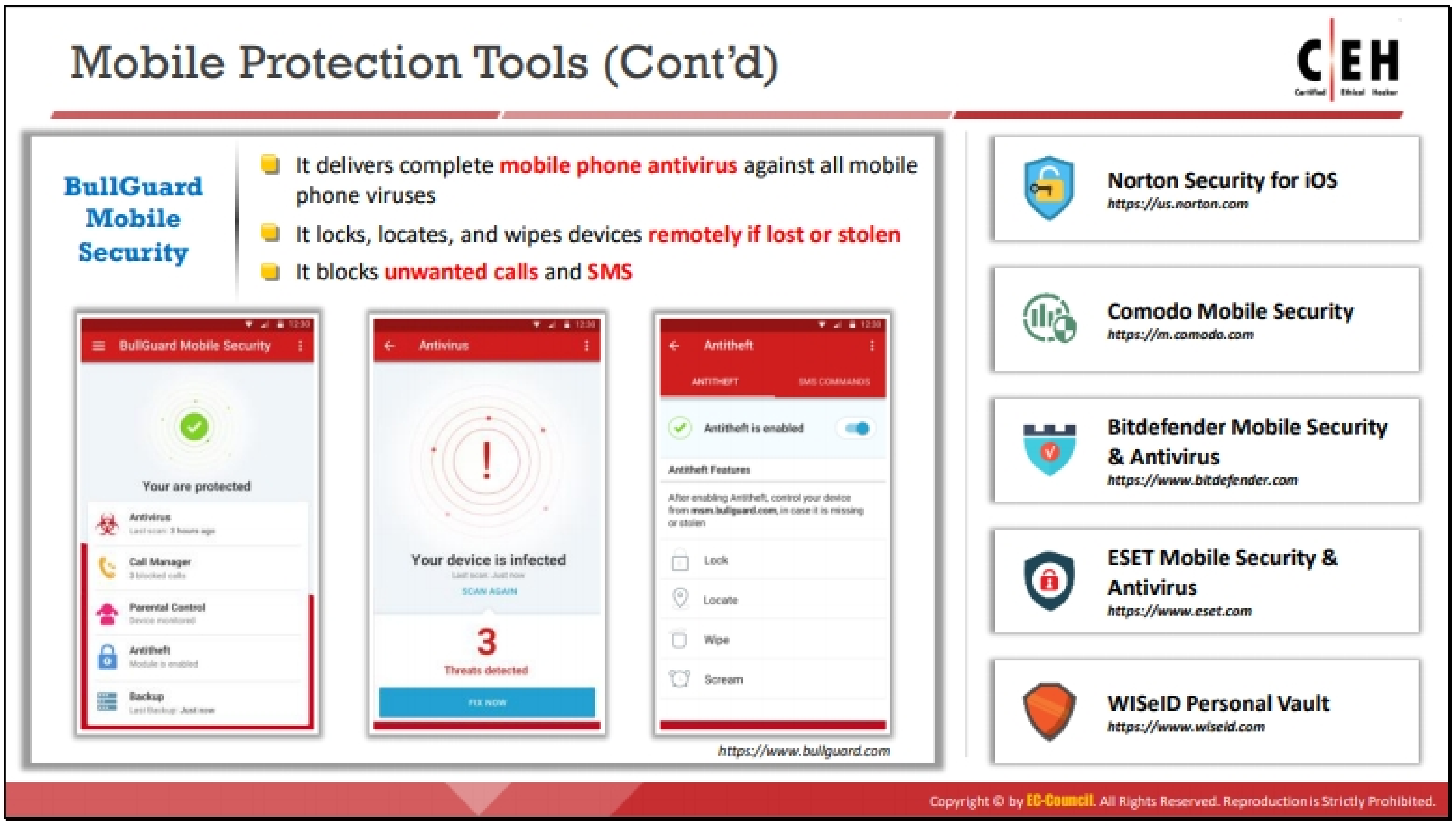
https://www.lookout.com

Zimperium's zIPS

- Zimperium's zIPS is a mobile intrusion prevention system app that provides comprehensive protection for iOS and Android devices against mobile network, device, and application cyber attacks
- It can detect both known and unknown threats by analyzing the behavior of your device

3:34 PM
zIPS
Threat Detected!
5 Active Issues
Device Safety
Network Safety
Apps Safety
Activity Monitor

https://www.zimperium.com



Mobile Protection Tools (Cont'd)

BullGuard Mobile Security

- It delivers complete mobile phone antivirus against all mobile phone viruses
- It locks, locates, and wipes devices remotely if lost or stolen
- It blocks unwanted calls and SMS

BullGuard Mobile Security
Your are protected
Antivirus
Call Manager
Parental Control
Antitheft
Backup
Antivirus
Your device is infected
SCAN AGAIN
3
Threats detected
FIX NOW
Antitheft
ANTITHEFT
SMS COMMANDS
Antitheft is enabled
Antitheft Features
Lock
Locate
Wipe
Scream

https://www.bullguard.com

Norton Security for iOS
https://us.norton.com

Comodo Mobile Security
https://m.comodo.com

Bitdefender Mobile Security & Antivirus
https://www.bitdefender.com

ESET Mobile Security & Antivirus
https://www.eset.com

WISeID Personal Vault
https://www.wiseid.com

Mobile Protection Tools

- **Lookout Personal**

  Source: *https://www.lookout.com*

  Lookout Personal helps to protect your device from security threats, loss, and theft. It is available for Android and iOS devices. It provides mobile security, identity protection, and theft prevention in a single app.

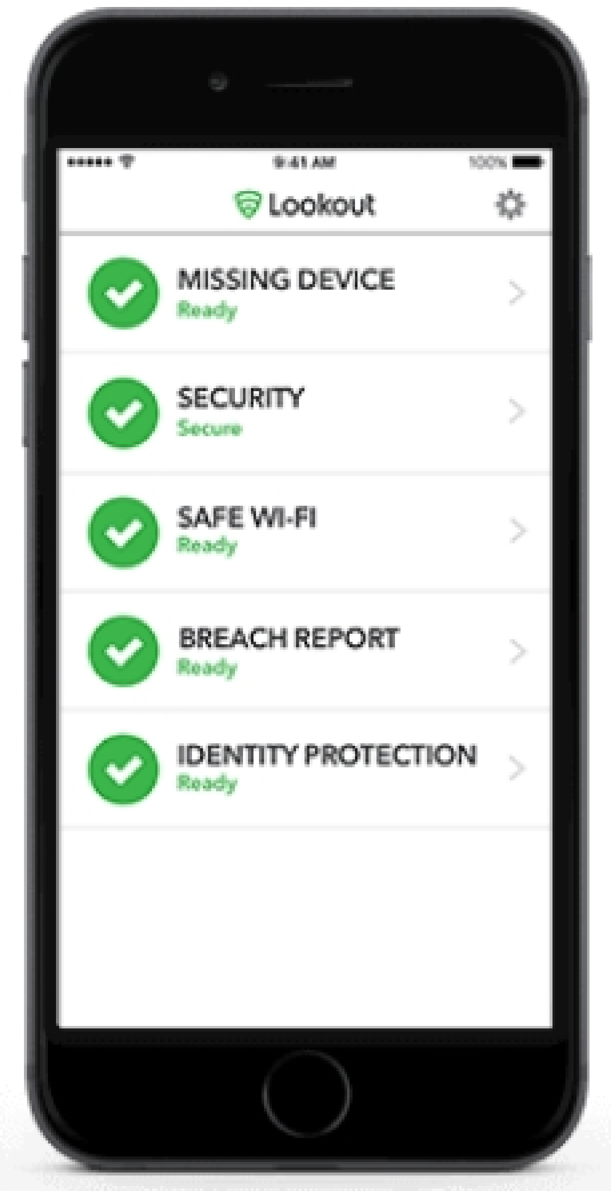


Figure 17.71: Screenshot of Lookout Personal

- **Zimperium's zIPS**

  Source: *https://www.zimperium.com*

  Zimperium's zIPS is a mobile intrusion prevention app that provides comprehensive protection for iOS and Android devices against mobile network, device, and application cyber-attacks. It uses advanced machine-learning techniques to identify and prevent both network-based and host-based threats such as

  - MITM attacks that can intercept your passwords and other confidential information when you are using public or private Wi-Fi networks
  - SpearPhishing attacks that can compromise high-value targets in your organization and infect them with data-stealing code
  - Reconnaissance scans that identify APTs and compromised devices in your network
  - Rogue Wi-Fi AP attacks that can hijack secure SSL sessions to steal confidential information

  zIPS is equipped with a behavioral analysis engine to automatically detect and block malicious threats by monitoring how they change the characteristics of the mobile device. It scans all mobile applications and browsers to enhance the security of your device and protects your entire organization from MITM, IPv4, and IPv6 attacks. It provides automated alerts to both the security officer and the user in the event of an incident. Furthermore, it uses "nonintrusive packet monitoring" to detect advanced mobile threats.

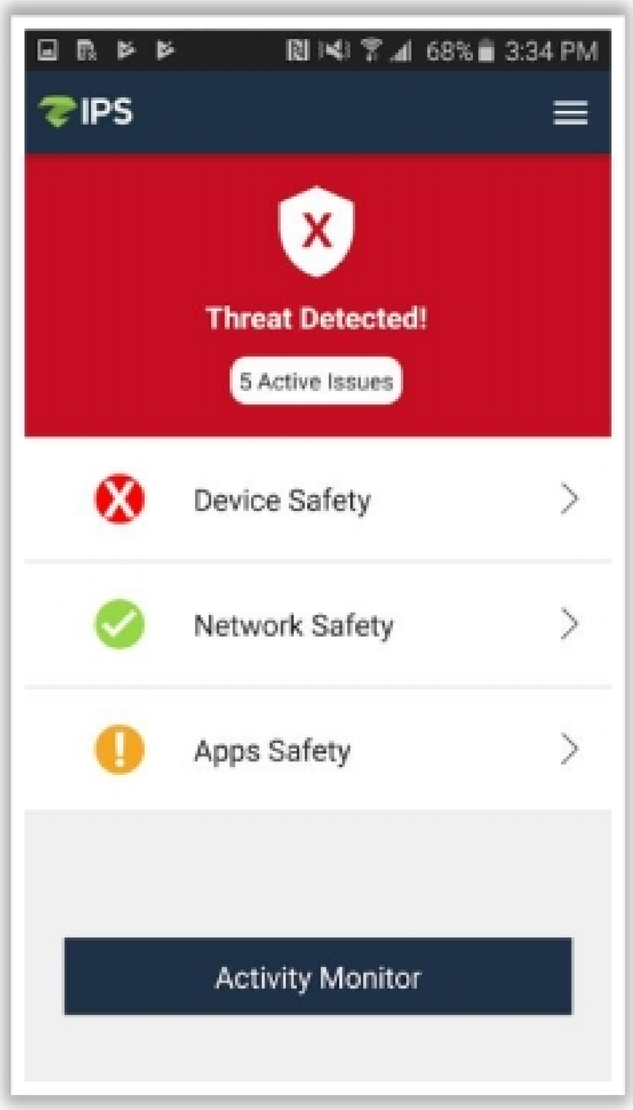
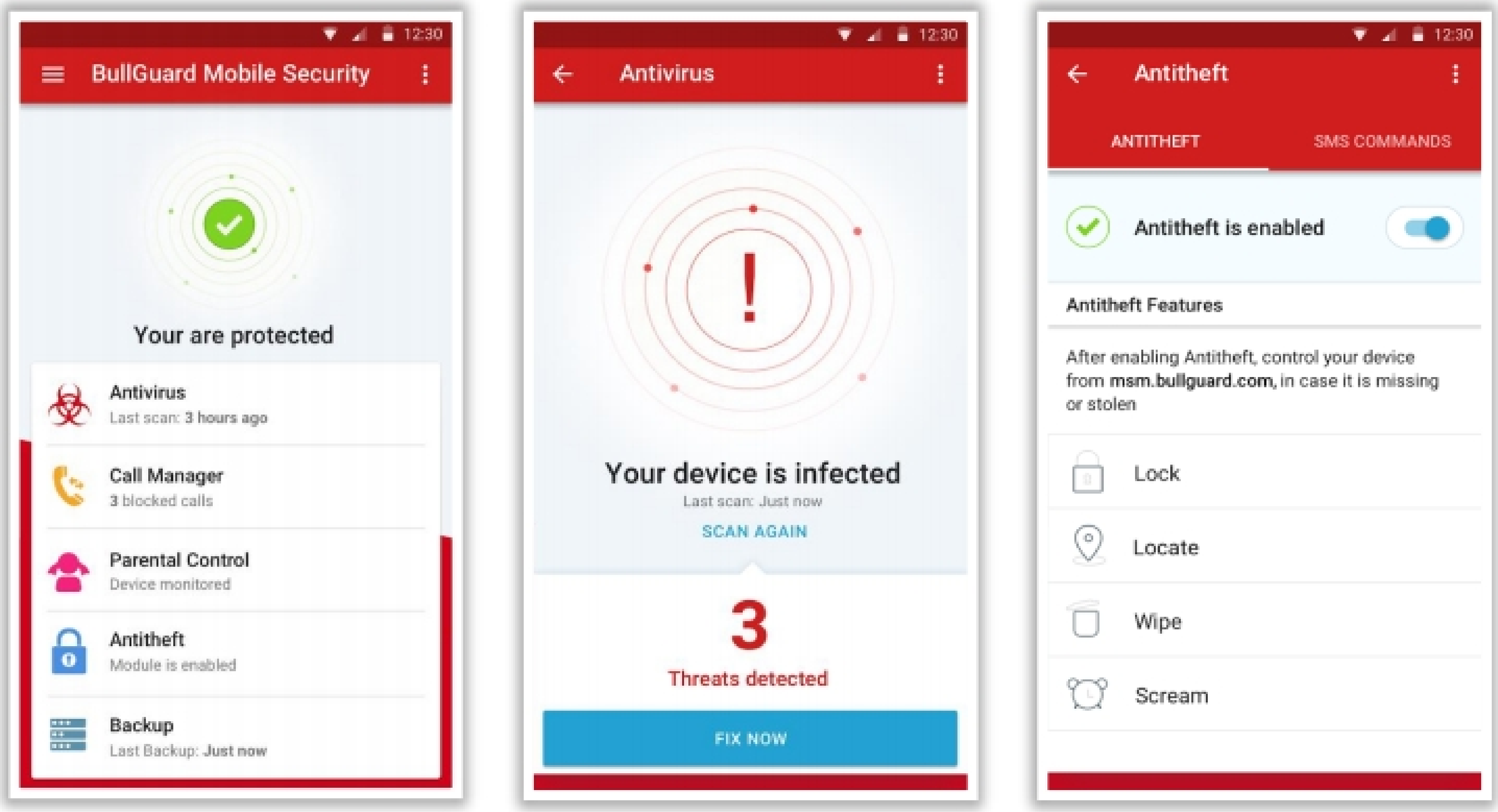

68% 3:34 PM
IPS
Threat Detected!
5 Active Issues
Device Safety
Network Safety
Apps Safety
Activity Monitor

Figure 17.72: Screenshot of Zimperium's zIPS

- **BullGuard Mobile Security**

  Source: *https://www.bullguard.com*

  BullGuard Mobile Security is an app for Android devices that provides total protection of the devices and personal data. It provides complete mobile phone anti-virus protection against all mobile phone viruses. It locks, locates, and wipes data remotely if the device is lost or stolen. It also blocks unwanted calls and SMS.

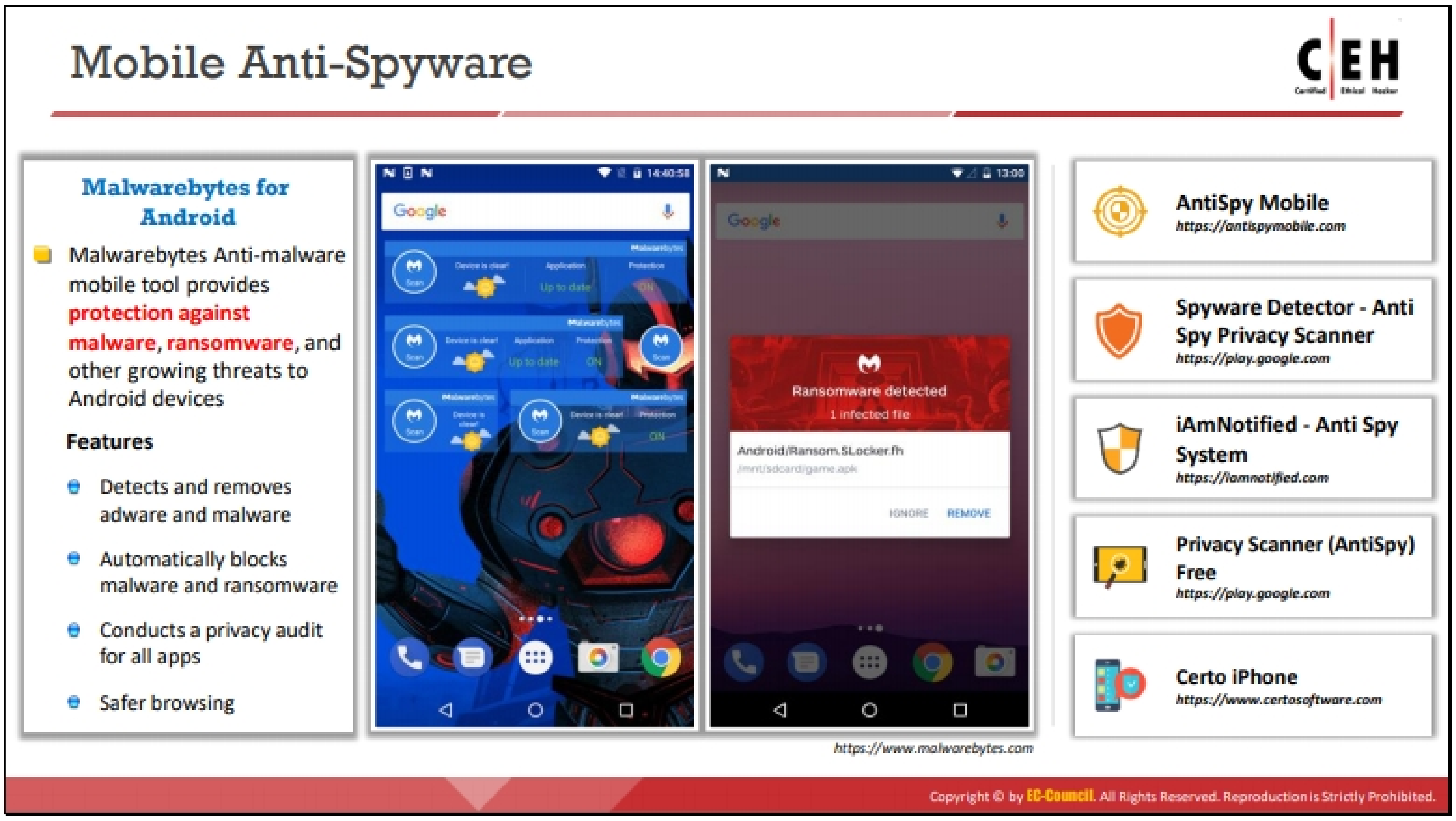

12:30
BullGuard Mobile Security
Your are protected
Antivirus
Last scan: 3 hours ago
Call Manager
3 blocked calls
Parental Control
Device monitored
Antitheft
Module is enabled
Backup
Last Backup: Just now
12:30
Antivirus
Your device is infected
Last scan: Just now
SCAN AGAIN
3
Threats detected
FIX NOW
12:30
Antitheft
ANTITHEFT
SMS COMMANDS
Antitheft is enabled
Antitheft Features
Lock
Locate
Wipe
Scream

After enabling Antitheft, control your device from **msm.bullguard.com**, in case it is missing or stolen

Figure 17.73: Screenshots of BullGuard Mobile Security

Some additional mobile protection tools are as follows:

- Norton Security for iOS (*https://us.norton.com*)
- Comodo Mobile Security (*https://m.comodo.com*)
- Bitdefender Mobile Security & Antivirus (*https://www.bitdefender.com*)
- ESET Mobile Security & Antivirus (*https://www.eset.com*)
- WISeID Personal Vault (*https://www.wiseid.com*)



Mobile Anti-Spyware

Malwarebytes for Android

- Malwarebytes Anti-malware mobile tool provides **protection against malware**, **ransomware**, and other growing threats to Android devices

Features

- Detects and removes adware and malware
- Automatically blocks malware and ransomware
- Conducts a privacy audit for all apps
- Safer browsing

https://www.malwarebytes.com

AntiSpy Mobile
https://antispymobile.com

Spyware Detector - Anti Spy Privacy Scanner
https://play.google.com

iAmNotified - Anti Spy System
https://iamnotified.com

Privacy Scanner (AntiSpy) Free
https://play.google.com

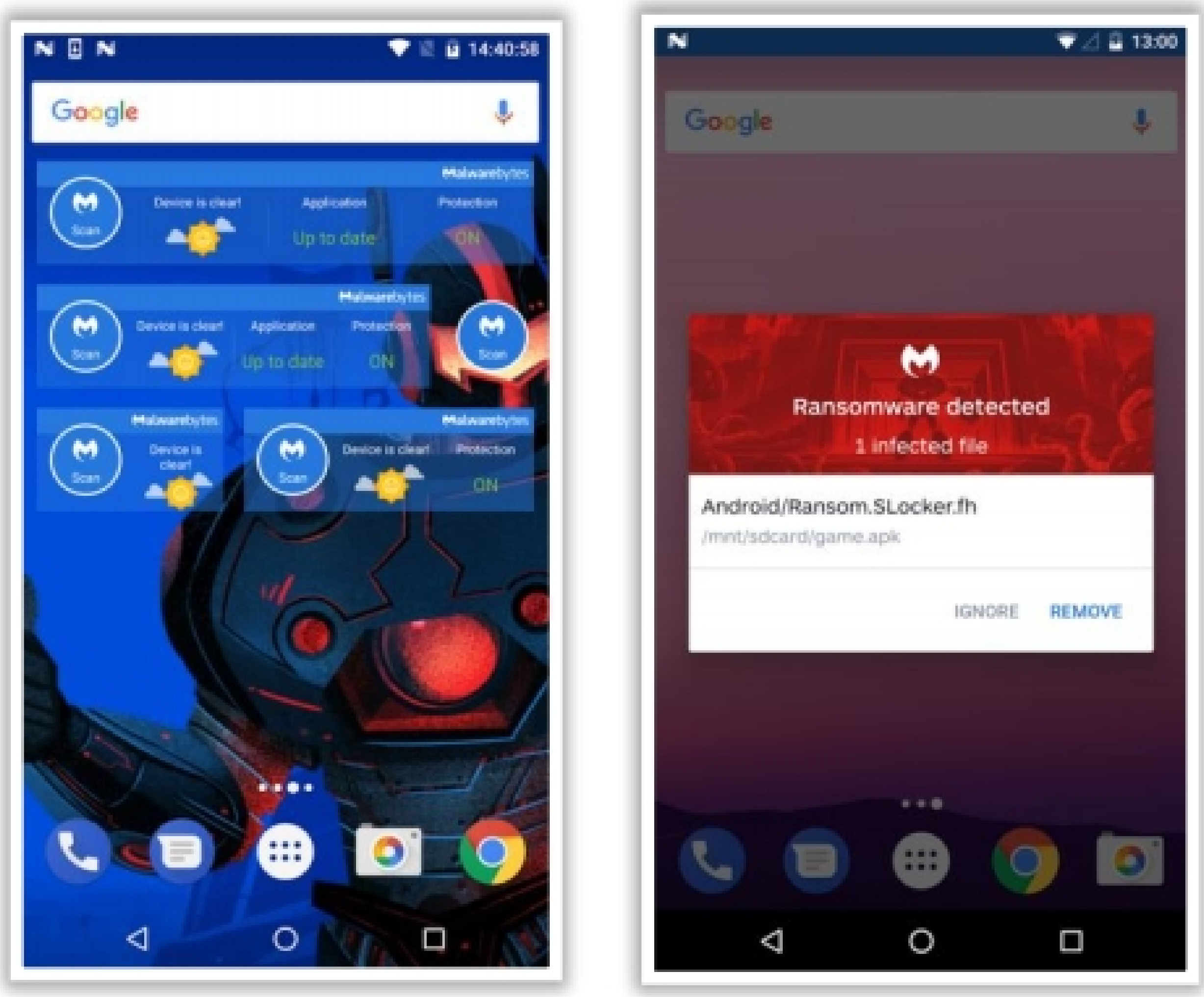
Certo iPhone
https://www.certosoftware.com

Mobile Anti-Spyware

- **Malwarebytes for Android**

  Source: *https://www.malwarebytes.com*

  Malwarebytes is an anti-malware mobile tool that provides protection against malware, ransomware, and other growing threats to Android devices. It detects and removes adware and malware, blocks malware and ransomware automatically, conducts privacy audits for all apps, and provides safe browsing.



Ransomware detected
1 infected file
Android/Ransom.SLocker.fh
/mnt/sdcard/game.apk

Figure 17.74: Screenshot of Malwarebytes Anti-Malware

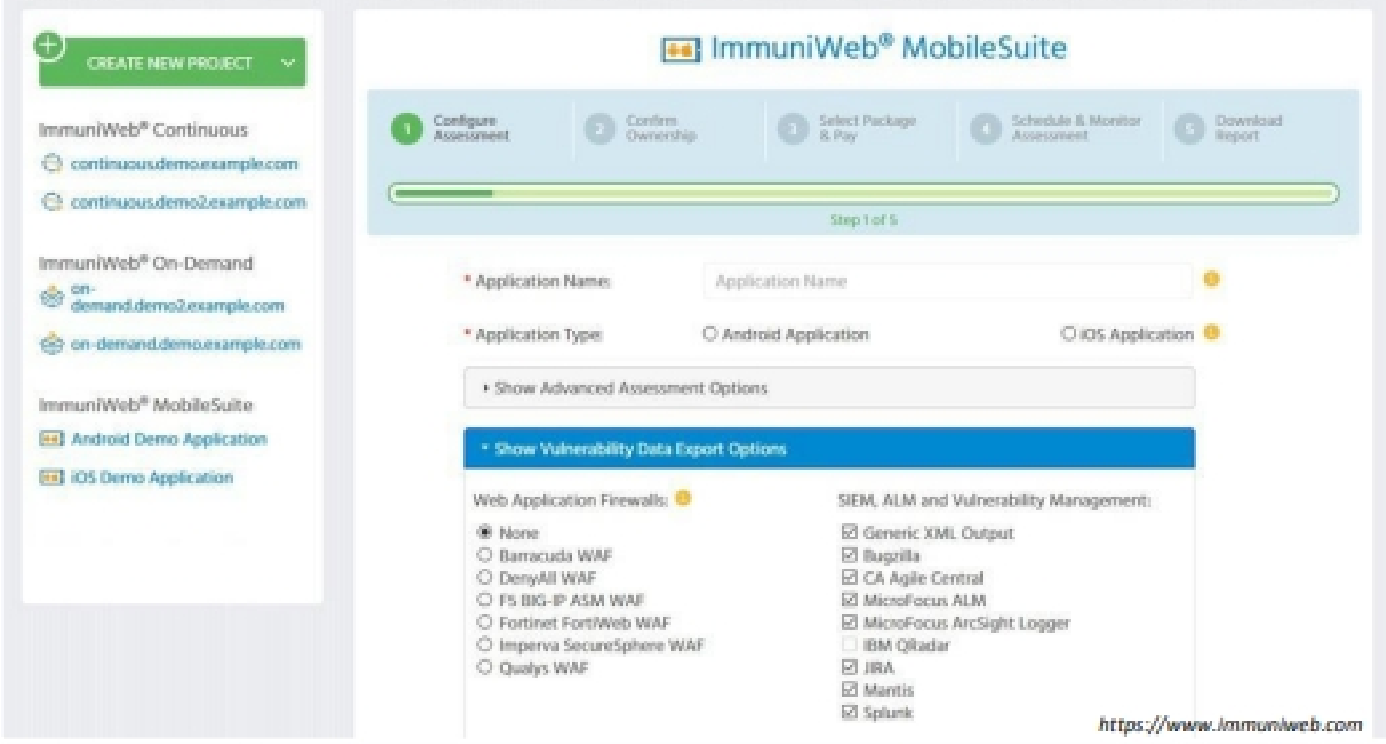
Some additional mobile anti-spyware tools are as follows:

- AntiSpy Mobile (*https://antispymobile.com*)
- Spyware Detector - Anti Spy Privacy Scanner (*https://play.google.com*)
- iAmNotified - Anti Spy System (*https://iamnotified.com*)
- Privacy Scanner (AntiSpy) Free (*https://play.google.com*)
- Certo iPhone (*https://www.certosoftware.com*)

Mobile Pen Testing Toolkit: ImmuniWeb® MobileSuite

- ImmuniWeb® MobileSuite leverages **Machine Learning technology** to augment and accelerate manual mobile penetration testing of iOS and Android mobile applications

- **Zero False-Positive SLA**
- **Threat-Aware Risk Scoring**
- **Tailored Remediation Guidelines**
- **CVE, CWE, and CVSSv3 scores**


CREATE NEW PROJECT
ImmuniWeb® Continuous
continuous.demo.example.com
continuous.demo2.example.com
ImmuniWeb® On-Demand
on-demand.demo2.example.com
on-demand.demo.example.com
ImmuniWeb® MobileSuite
Android Demo Application
iOS Demo Application
ImmuniWeb® MobileSuite
1 Configure Assessment
2 Confirm Ownership
3 Select Package & Pay
4 Schedule & Monitor Assessment
5 Download Report
Step 1 of 5
▸ Show Advanced Assessment Options
▾ Show Vulnerability Data Export Options

* Application Name: Application Name

* Application Type: ○ Android Application ○ iOS Application

Web Application Firewalls:
● None
○ Barracuda WAF
○ DenyAll WAF
○ F5 BIG-IP ASM WAF
○ Fortinet FortiWeb WAF
○ Imperva SecureSphere WAF
○ Qualys WAF

SIEM, ALM and Vulnerability Management:
☑ Generic XML Output
☑ Bugzilla
☑ CA Agile Central
☑ MicroFocus ALM
☑ MicroFocus ArcSight Logger
☐ IBM QRadar
☑ JIRA
☑ Mantis
☑ Splunk

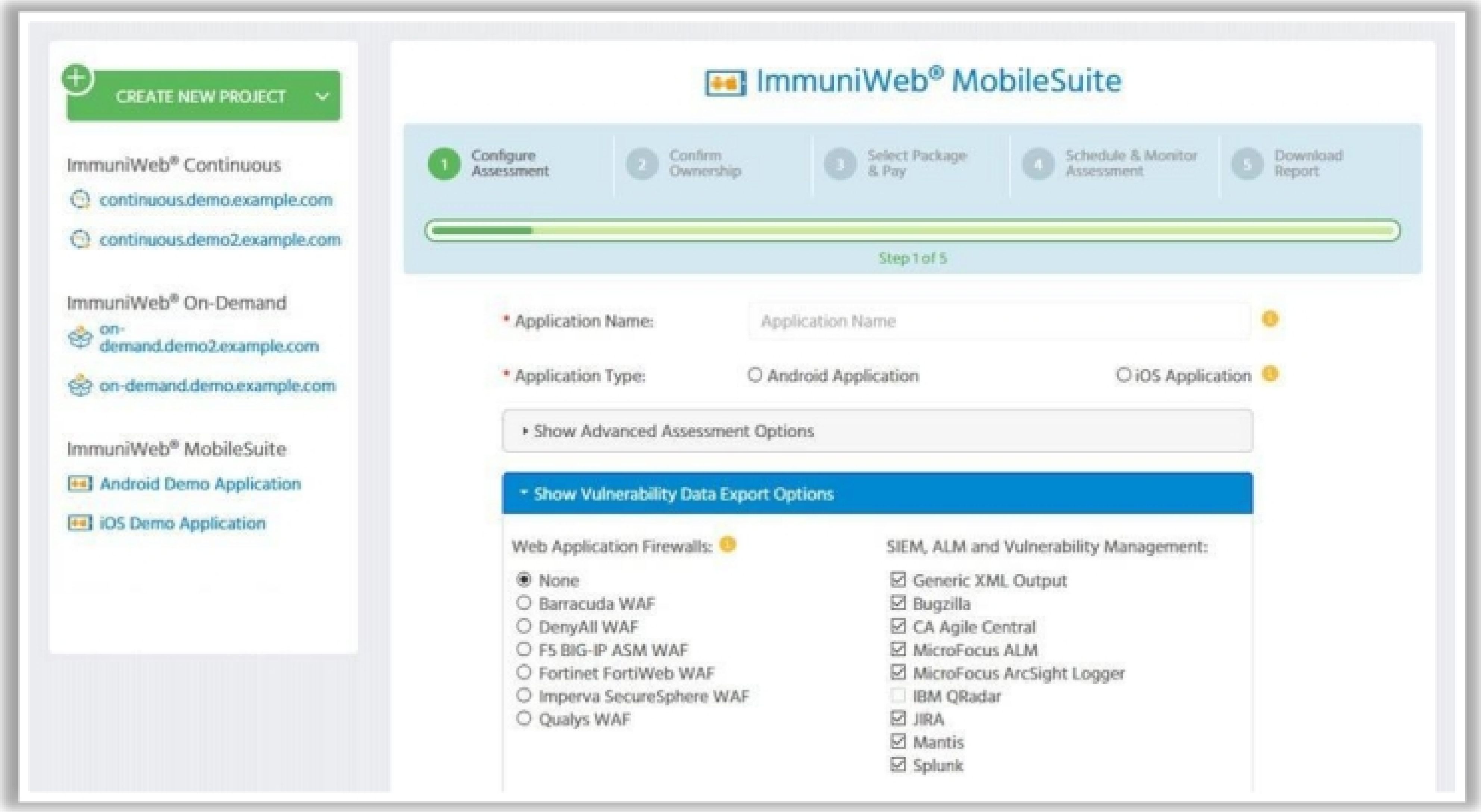
https://www.immuniweb.com

Mobile Pen Testing Toolkit

- **ImmuniWeb® MobileSuite**

  Source: *https://www.immuniweb.com*

  ImmuniWeb® MobileSuite leverages machine learning technology to augment and accelerate manual mobile penetration testing of iOS and Android mobile applications. It provides scalable, rapid, and DevSecOps-enabled mobile app and backend testing with tailored remediation guidelines and zero false-positives SLA. Furthermore, it provides SDLC and CI/CD tools integration and WAF for mobile backend flaws. Using this toolkit, security professionals can perform static, dynamic, and interactive security testing with SCA. It also provides various reports such as Threat-Aware Risk Scoring, Tailored Remediation Guidelines, CVE, CWE and CVSSv3 scores.


CREATE NEW PROJECT
ImmuniWeb® Continuous
continuous.demo.example.com
continuous.demo2.example.com
ImmuniWeb® On-Demand
on-demand.demo2.example.com
on-demand.demo.example.com
ImmuniWeb® MobileSuite
Android Demo Application
iOS Demo Application
ImmuniWeb® MobileSuite
1 Configure Assessment
2 Confirm Ownership
3 Select Package & Pay
4 Schedule & Monitor Assessment
5 Download Report
Step 1 of 5
Show Advanced Assessment Options
Show Vulnerability Data Export Options

* Application Name: Application Name

* Application Type: ○ Android Application ○ iOS Application

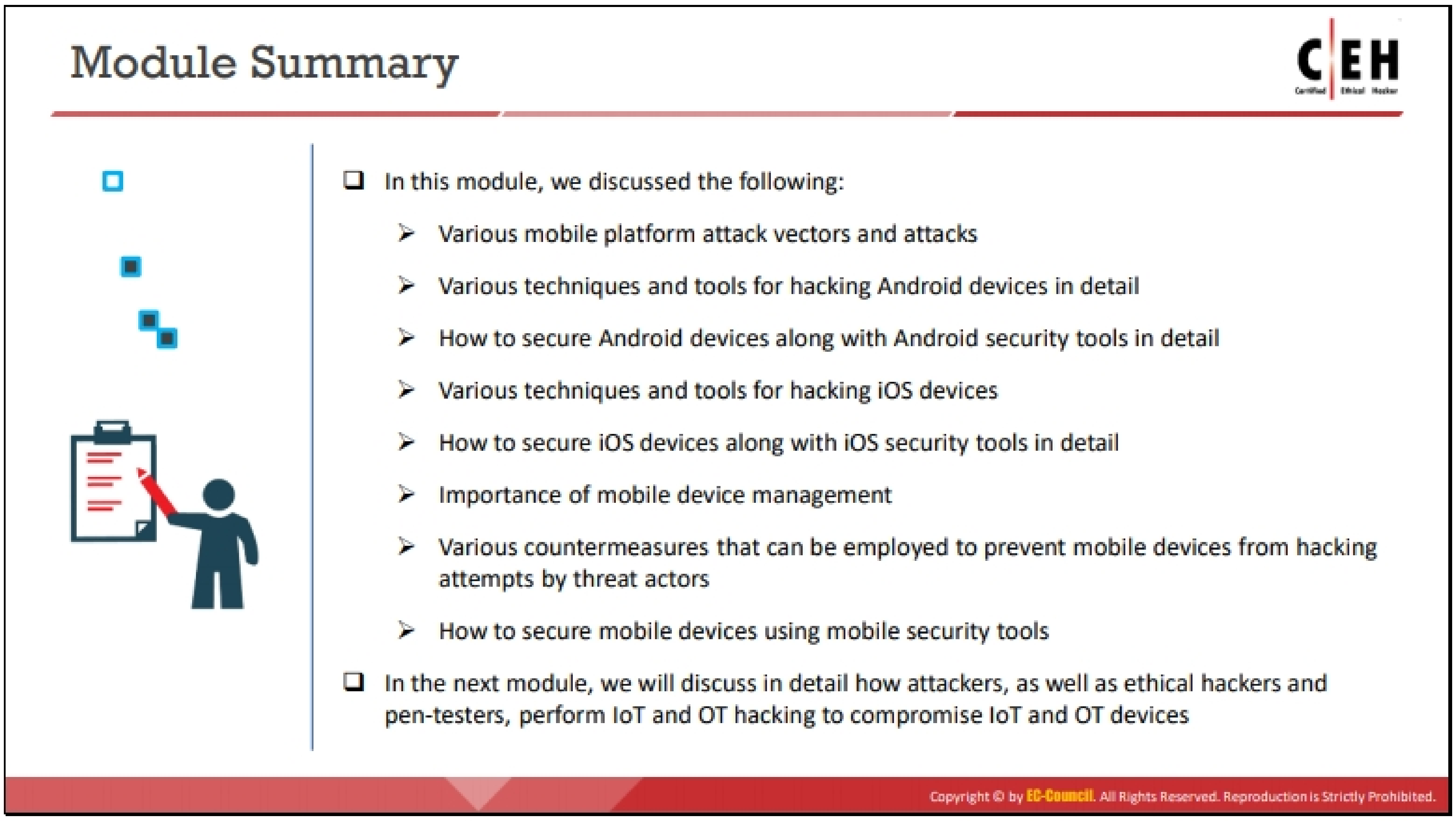
Web Application Firewalls:

● None
○ Barracuda WAF
○ DenyAll WAF
○ F5 BIG-IP ASM WAF
○ Fortinet FortiWeb WAF
○ Imperva SecureSphere WAF
○ Qualys WAF

SIEM, ALM and Vulnerability Management:

☑ Generic XML Output
☑ Bugzilla
☑ CA Agile Central
☑ MicroFocus ALM
☑ MicroFocus ArcSight Logger
☐ IBM QRadar
☑ JIRA
☑ Mantis
☑ Splunk

Figure 17.75: Screenshots of ImmuniWeb® MobileSuite



Module Summary

- ☐ In this module, we discussed the following:
    - ➢ Various mobile platform attack vectors and attacks
    - ➢ Various techniques and tools for hacking Android devices in detail
    - ➢ How to secure Android devices along with Android security tools in detail
    - ➢ Various techniques and tools for hacking iOS devices
    - ➢ How to secure iOS devices along with iOS security tools in detail
    - ➢ Importance of mobile device management
    - ➢ Various countermeasures that can be employed to prevent mobile devices from hacking attempts by threat actors
    - ➢ How to secure mobile devices using mobile security tools
- ☐ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform IoT and OT hacking to compromise IoT and OT devices

Module Summary

This module discussed various mobile platform attack vectors and attacks. In addition, it discussed various techniques and tools for hacking Android devices in detail. It also provided a detailed explanation of how Android devices can be secured using Android security tools. Furthermore, it described various techniques and tools for hacking iOS devices. It also discussed how iOS devices can be secured using iOS security tools. Moreover, it emphasized the importance of mobile device management. Subsequently, it presented various countermeasures to protect mobile devices from hacking attempts by threat actors. Finally, it ended with a detailed discussion on how mobile devices can be secured using mobile security tools.

In the next module, we will discuss in detail how attackers as well as ethical hackers and pen-testers perform IoT and OT hacking to compromise IoT and OT devices.